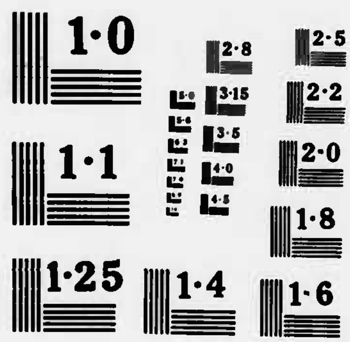
AD-A158 118 STRATEGIES FOR ASSOCIATING DATA AND LOCATION IN A GEOGRAPHIC INFORMATION SYSTEM(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH R G WAITON 1985 1/2

UNCLASSIFIED AFIT/CI/NR-85-78T F/G 8/2 NL



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

(1)

AD-A158 118

Strategies for Associating Data and Location in a Geographic Information System

Richard G. Walton, Capt, USAF

Comprehensive Project Report
Master of Science in Computer Science
University of California, Los Angeles
1985

Report: 78 Pages; Accompanying Annotated Bibliography: 21 Pages

ABSTRACT

Much of the existing work in the area of Geographic Information Systems (GIS) treats spatial objects, e.g. points, lines, and regions, as the primary entities of interest. In that approach, descriptive information is associated directly with each of these objects, and location is seen as being merely one of these data items. This paper explores the feasibility of implementing an alternative design which uses Location Data Sets and Location Predicates as the basic entities managed by a Location Data Management System (LDMS). A major advantage of the proposed approach is its suitability for automatic enforcement of data consistency across multi-scale geographic entities.

The central idea of the Location Data Set approach is that spatial data should be directly associated with locations rather than named regions or points. The relationships between geographic entities and data values may then be derived through the intermediate relationship of shared location. It is envisioned that each type of data which is distributive in nature would be stored in a separate set. Data values associated with conventional points, lines, and regions would then be merely restrictions on these global data sets. This is similar to the way in which the external views of a database represent a subsetting of the global data.

The paper includes a survey of fifteen selected GIS implementations and existing work relevant to identified implementation obstacles.

JW

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR 85-78T	2. GOVT ACCESSION NO. AD-A158 118	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Strategies for Associating Data and Location in a Geographic Information System		5. TYPE OF REPORT & PERIOD COVERED THESIS/DISSERTATION
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard G. Waiton		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: University of California Los Angeles		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS AFIT/NR WPAFB OH 45433		12. REPORT DATE 1985
		13. NUMBER OF PAGES 78
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASS
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

APPROVED FOR PUBLIC RELEASE: IAW AFR 190-1

Lynn Wolaver

LYNN E. WOLAVER
Dean for Research and
Professional Development
5 AUG 1985 AFIT, Wright-Patterson AFB OH

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASS

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

85 8 13 079

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to ascertain the value and/or contribution of research accomplished by students or faculty of the Air Force Institute of Technology (AU). It would be greatly appreciated if you would complete the following questionnaire and return it to:

AFIT/NR
Wright-Patterson AFB OH 45433

RESEARCH TITLE: Strategies for Associating Data and Location in a Geoxgraphic Information System

AUTHOR: Richard G. Waiton

RESEARCH ASSESSMENT QUESTIONS:

1. Did this research contribute to a current Air Force project?

( ) a. YES ( ) b. NO

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not?

( ) a. YES ( ) b. NO

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency achieved/received by virtue of AFIT performing the research. Can you estimate what this research would have cost if it had been accomplished under contract or if it had been done in-house in terms of manpower and/or dollars?

( ) a. MAN-YEARS ________ ( ) b. $________

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3. above), what is your estimate of its significance?

( ) a. HIGHLY SIGNIFICANT ( ) b. SIGNIFICANT ( ) c. SLIGHTLY SIGNIFICANT ( ) d. OF NO SIGNIFICANCE

5. AFIT welcomes any further comments you may have on the above questions, or any additional details concerning the current application, future potential, or other value of this research. Please use the bottom part of this questionnaire for your statement(s).

NAME ________ GRADE ________ POSITION ________

ORGANIZATION ________ LOCATION ________

STATEMENT(s):

Strategies for Associating Data and Location in a Geographic Information System

1. Introduction

The capabilities of geographic data processing systems have been been greatly expanded in recent years. In analogy with Management Information Systems, these enhanced spatial data management systems have become generally known as Geographic Information Systems (GISs). The fields of computer science that are most relevant to GIS design are computer graphics, image processing and pattern recognition, data structures, and database design.[32]

In general, any information which may be interpreted, manipulated, or referenced in a spatial context is a candidate for a geographic information system. The software trend has been toward greater user orientation and generality with increased interactive graphics capabilities. Early data input methods, which frequently involved manual encoding of data point by point, or hand measurement of feature coordinates, have given way to electromechanical and electronic digitizers and scanners.[9]

This paper will survey the more conventional approaches to GIS design and propose an alternative approach suitable for a wide range of geographically-

oriented applications. This alternative software design, while incorporating or extending features of some existing systems, differs considerably in its strategy for associating spatial data and location. While an actual implementation would need to overcome a number of major obstacles, a review of related work shows that none of these are insurmountable. The paper concludes with the outline of a workable initial implementation and a discussion of possible future extensions.

2. Intended Application Area

The type of geographic information system proposed here does not fit conveniently into any of the major subclassifications of such systems. It does, however, seek to incorporate many of the desirable features of existing geographic information systems as well as some features of conventional database management systems. Where it most differs is in the area of data integrity constraints. A primary goal is to develop the means of enforcing spatial data consistency and semantic constraints, an area not well developed in most existing systems. Additionally, it will be capable of managing data relevant to any region of the world without intrinsic limitations of scale or resolution. The result might best be described as a generalized and integrated locational data filing, manipulation, and retrieval facility combined with interactive graphic display features. In the interest of brevity, it will be hereafter referred to as a Location Data Management System (LDMS).

While LDMS will be capable of managing alphanumeric data, manipulation of such data would be limited to operations defined in a spatial context. The system is not intended to replace a full-featured conventional database, but rather to

function as an independent or cooperating system for the management of geographic data.

Data output will be primarily in the form of maps, with emphasis on an interactive graphics format rather than the conventional printed product. The user will be able to define geographic entities of interest and then request display of one or more of them by name. The system will select an appropriate scale to display requested entities within the context of their surrounding region. The user may then directly determine any relationships of interest, or indicate specific positions on the screen and pose queries in terms of these.

LDMS will include many of the features of land use analysis and planning systems. Indeed, one of its main goals is to maintain the integrity and consistency of the types of data managed by such systems. Some specific potential application possibilities for the proposed system include the following:

Scale-independent file system. Travel agencies, news organizations, and intelligence-gathering bureaus would be potential users here. Reports dealing with geographic locations could be maintained in an independent text storage and retrieval system. References to those reports would then be entered into LDMS at the highest applicable resolution level. The references could take the form of file names (conventional or computer), publication name and issue, internal report numbers, or pointers into a cooperating system (e.g. videodisc track numbers). A tourist, for example, could interactively identify his route of travel to select information on enroute attractions.

Regional planning and resource management. The scale independence and data consistency enforcement features of the system are designed specifically with such uses in mind. They ensure that data may be entered and retrieved at any resolution level.

Marketing and site location analysis. Data relevant to such matters could be maintained by LDMS. Selection criteria would then be applied to the data to produce an output display of locations satisfying those criteria. The user could declare the resulting locations as newly-defined geographic entities, and exploit the interactive features of the system to perform further analysis on them.

3. The Conventional Approach

There are some valid differences of opinion as to what constitutes a GIS. One of the major problems in the field is that it includes a variety of supporting disciplines and many of these use different jargon and assumptions.[32] Thus, the nature of the data handled, and the types of logical operations which may be performed on it, vary greatly from one GIS to the next. In general, however, stored data can be viewed as being either *pictorial, descriptive,* or *semantic*. Most present systems are application-specific and therefore tend to focus on operations involving only one or two of these information types.

Manipulation of *pictorial* data involves storage and retrieval of visual display information, without regard to its meaning; the system views it as merely a structured collection of graphics patterns. Cartographic systems typically support

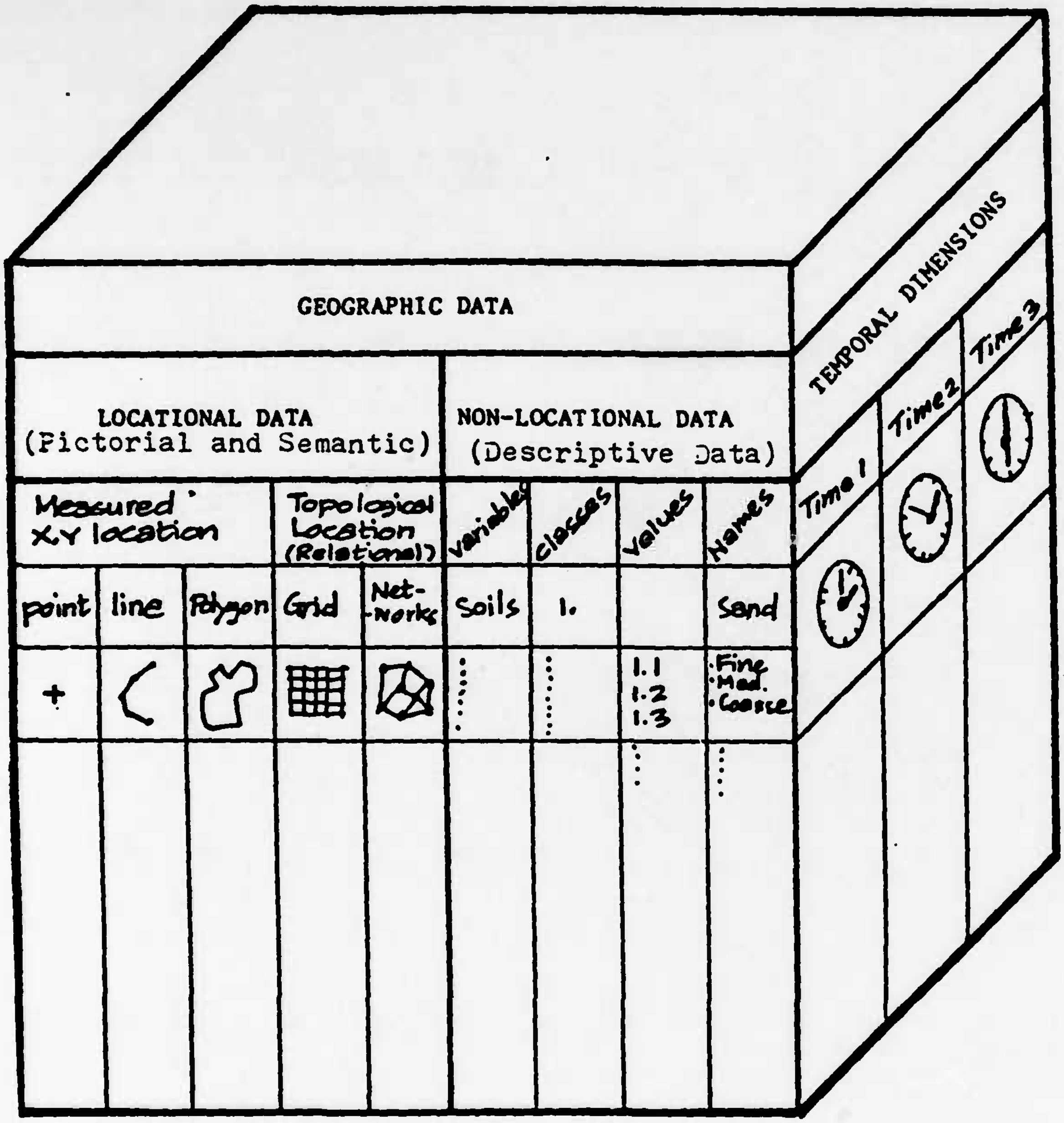
highly specialized operations on pictorial data.

Descriptive information is the type of information managed by conventional DBMS's. This includes attributes and those selected relationships which have been explicitly stored by the user. For example, 1984 rainfall in Los Angeles or that city's distance from each world capital. One could even go so far as to explicitly store whole hierarchies of geographic subdivisions by name, but that would not by itself produce a true GIS.[29]

Semantic information refers to relationships and associations which have not been separately enumerated and stored, but which follow from the spatial characteristics of stored geographic entities. Distances between geographical entities and containment of one entity by another fall in this category. Operations upon data semantics are typically implemented as limited sets of pre-defined functions and are frequently application specific.

An alternative classification scheme is to divide the previous three information types into geometric and non-geometric attributes classes. That analysis considers location and shape to be geometric attributes and non-geometric attributes to be either nominal (category-defining), scalar, or some combination of these; "River", elevation, and "population density" are respective examples.[29,25]

While application program independence from physical data structuring considerations is well developed for conventional DBMSs, that is not the case with geographic applications. [29] In part, this is because manipulation of spatial data poses problems that conventional DBMSs need not address. Many of these are due to the fact that manipulations of spatial data frequently result in values for



THREE CONCEPTUAL COMPONENTS OF A GEOGRAPHIC BASED INFORMATION SYSTEM

Note the impact of the temporal dimension upon data volume. Geographic information systems designed to manage remote-sensed data are especially affected.

Source: Dangermond, "A Classification of Software Components Commonly Used in Geographic Information Systems"

relationships rather than entity names or attribute values.[34]

3.1. General Design Philosophy

Systems differ greatly in their ability to handle both semantic and pictorial aspects of geographic data. Many are rigid and do not allow the user to add needed functions or data categories not foreseen by the system designer. WRIS, ODYSSEY, and KANDIDATS, for example, must rewrite the entire data base to add an image.[5] Not all were even originally conceived and designed specifically as geographic information systems; AGS, GADS, GEO-QUEL, and IMAID were either defined as conventional database management systems or designed as extensions to them.

Underlying design philosophies generally follow one of two patterns:

1. Creation of an application-specific GIS from the ground up, perhaps taking advantage of certain commercially available statistical or report generation packages. Many of these systems consist of only a handful of FORTRAN programs.[29] A common practice is to use a minicomputer to perform data entry and pre-processing, and a larger mainframe to provide data manipulation.
2. Superposition of a more general-purpose GIS upon an underlying conventional DBMS. The fit between the two often appears forced due to the inability of the underlying DBMS to efficiently deal with the nature and quantity of spatial data.

As regards access structure design, three major trends exist:

1. Separate the total area into regions based on their location in a reference coordinate system. Individual records or attribute data are clustered with each region.
2. Group the attribute data and spatial entities separately and provide indices or other means for associating the two. Pictorial and attribute data may be resident on different types of storage media, supported by special-purpose hardware to perform data selection on each.
3. Treat all spatial objects as wholly independent entities, perhaps grouping them on the basis of some shared characteristic. This is the approach adopted by several systems implemented as extensions to conventional DBMSs.

Examples of Existing Geographic Information and Cartographic Systems

SYSTEM	DEFINITION	OPERATOR
AGS	Amoco Graphics System	Amoco
BASIS	Bay Area Spatial Information System	Association of Bay Area Governments
CGIS	Canada Geographical Information System	Dept. of the Environment, Canada
GADS	Geo-Data Analysis and Display System	IBM
GEO-QUEL	Geographical Extension of QUEL (INGRES)	University of California, Berkeley
IBIS	Image-Based Information System	Purdue University
IMAID	Integrated Image Analysis and Image Database Mgmnt Sys	Jet Propulsion Laboratories
KANDIDATS	Kansas Digital Image Data System	University of Kansas
ODYSSEY, POLYVRT		Harvard University
PICDMS	Pictorial Database Management System	University of California, Los Angeles
REDI (QPE)	Relational Database System for Images	Purdue University
STANDARD	Storage and Access of Network Data for Rivers and Drainage Basins	University of Nebraska
SYMAP	Synagraphic Mapping System	Northwestern Technological Institute
WRIS	Wildlife Resource Information System	US Forest Service

Source: adapted from Chock, "Manipulating Data Structures in Pictorial Information Systems"

The geographic information itself may be organized on auxiliary storage either as a *databank* or as a *database*. Databanks serve merely as repositories of information and offer simplicity, often at the cost of inflexibility. Complex operations are typically not supported and it is common to store data as a collection of sequential files, each of which contains the total data for a large region. A true database, on the other hand, is integration oriented and stores data on inter-entity relationships as well as the entities themselves.[29]

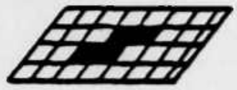
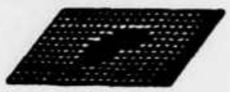
3.2. Structural Data Models

Although geographic information systems use only two major types of internal data organization, they go by a variety of names. We will generally refer to these two structural categories as *topological* and *grid*. Most systems elect to implement one or the other but not both.

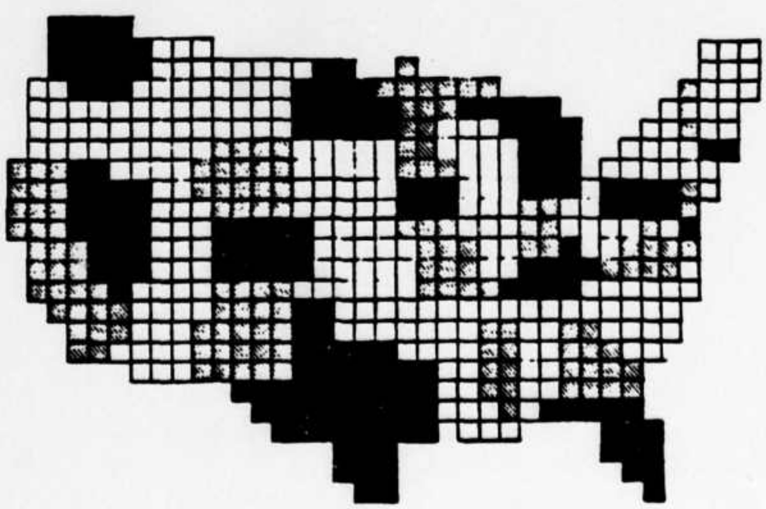
Topological systems take advantage of the convenient division of geographical entities into point, line, and region types. The terms *vector, linked,* or *polygon* format are also often used. In most cases, point entities are specified directly as coordinate pairs, with lines represented as chains of points. Regions are similarly defined in terms of the lines which form their boundaries. GEO-QUEL and IMAID, for example, store information in the form of points, line segments, and point pairs; GADS, WRIS, STANDARD, and CGIS maintain closed lists of points defining polygon regions. Some systems which use this approach store location and descriptive information separately and connect the two through elaborate pointer structures.[29]

Perhaps the greatest advantage of the topological format is that it is highly

APPROACHES TO GEO-CODING

METHOD	COST CONSIDERATION	OVERALL FLEXIBILITY
GRID CELL: 	MANUALLY OPERATED	SPATIAL RESOLUTION POOR, UPDATING DIFFICULT
POLYGON: 	EXPENSIVE FOR LARGE DATA SETS	CERTAIN OPERATIONS PROHIBITED
IMAGE RASTER: 	REQUIRES IMAGE PROCESSING TECHNOLOGY	NEITHER SCALE NOR DATA FORMAT DEPENDENT

Cartographic polygons.



Gridded polygons.

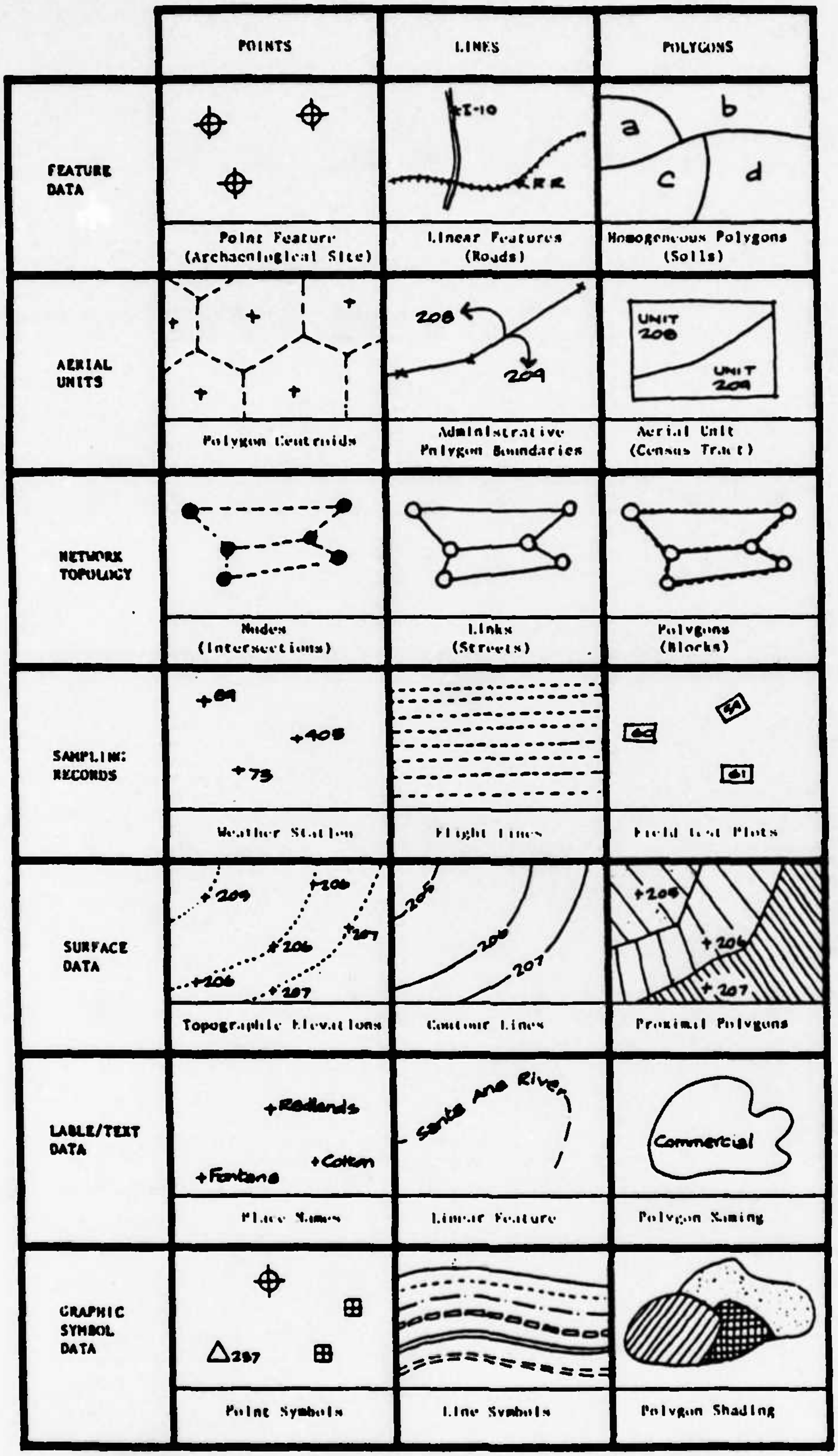
Sources: Zobrist et al, "Use of Landsat Imagery for Urban Analysis"
Chock et al, "Database Structure and Manipulation
Capabilities of a Picture Database Management System"

storage efficient. This advantage may be further increased through the use of suitable compression techniques. *Chain encoding*, for example, represents each unit increment of the coordinate point string by a single digit denoting its direction. *Delta encoding* approximates curves by converting them to a series of straight line segments whose length varies according to the local radius of curvature. [29] Topological structures are especially suitable for storing spatial objects for which sharp boundaries exist or can be imposed, such as political and legal subdivisions. Another advantage lies in the ability to directly apply many graph processing algorithms to data stored in this form. These advantages are responsible for making topological polygons by far the most common data structure used by pictorial systems.[3]

Unfortunately, the topological format also possesses several offsetting disadvantages. First, much of the data being collected today by remote sensing and other advanced technologies is in grid format.[33] Secondly, software is generally more complex than that based on the grid model, especially as regards data editing and update. Topological representations are a particularly poor choice for performing set algebra and distance-related operations. [29]

The *grid* approach superimposes a rectangular grid over the area of interest and associates each grid cell with one or more data records. For this reason, it is also sometimes known as *cellular* format. The values associated with each grid cell may represent either pixel intensity or any scalar or nominal data value associated with the cell coordinates. In general, the number and types of data fields are fixed, although some may be reserved for work space or planned expansions. Grid systems may be further subdivided into three types:

Options For Topological Data Representation



BREAKDOWN OF GEOGRAPHIC DATA TYPES & METHODS OF REPRESENTATION

Source: Dangermond, "A Classification of Software Components Commonly Used in Geographic Information Systems"

1. *Raster image systems,* in which each record is stored as a row of pixels.
2. *Matrix systems,* which represent pixel values by elements in a large two-dimensional array.
3. *Flat file systems,* which maintain a file record for each cell.[3]

One of the strengths of the grid method is its ability to represent transition information. Whereas topological structures define objects in terms of their boundaries, a high-resolution grid can assign one of a range of values to each cell in a transition area.

A drawback common to many grid systems is that a maximum resolution limit is fixed *a priori* by the software design. They also tend to be relatively storage intensive, and this imposes practical limits on the area of coverage, the resolution, or both of these. Major systems based on the grid model are limited to BASIS, IBIS, KANDIDATS, PICDMS, SYMAP, NORMAP, and Stanford Research Institute's HAWKEYE. [5]

3.3. Major Application Categories

Despite the breadth of potential application areas for geographic information systems, most existing ones have concentrated on one of two major areas: computer-assisted cartography and land use planning. The influence of these two disciplines on GIS development is not, however, universally regarded as a positive force. One recognized authority has charged that the three major supporting disciplines--geodesy and photogrammetry, computer science, and geography--failed to recognize their opportunity for leadership in the GIS field. Consequently, it fell by default to those disciplines least able to develop the necessary theoretical and quantitative underpinnings. Cartographers are especially faulted

Some Common Geographic Information System Application Areas

- SELECTED RELATIONSHIP BETWEEN RESPONSIBILITIES & GIS APPLICATIONS

Responsibilities	Examples of Data	GENERAL TASK FUNCTIONS SUPPORTED BY GEO. INFO.: Maintenance of Geographic Records (i.e., Operational & Inventory Data)	Graphic Display (Drafting)	Statistical Reporting	Planning	Management	Policy Making
Territorial Control	Boundary Survey	O	O				
Natural Resource Exploitation	Geophysical, Topo, Geology, Vegetation, Soils	O	+		O		
Taxation & Ownership Monitoring	Cadastral Surveys & Tax Records	O	O	+			
Land Use/ Infrastructure Planning	Opportunities & Constraint Data				O		
Land Use Zoning	Zoning Records	O	+		O		
Land Use/ Infrastructure Design/ Construction	Engineering Surveys/ Data		O		+		
Construction Facility Record Keeping & Management	As Built Drawings of Utilities, Buildings, etc.	O	O		+		
Development Measurement	Land Use Survey	+	+	O	+	O	O
Census of Statistics	Population, Housing, Health & Economic Data		+	O	O		O
Event Monitoring	Police/Fire Statistics		+	+	+	O	O
Natural Resource Management	Ongoing Forest Record Keeping	O	O		O	O	
Monitoring of Environment	Wildlife, Vegetation, Air, Soil, Water, etc.	O	+	O	O		O

O=Major
+=Minor

Source: Dangermond, "A Classification of Software Components Commonly Used in Geographic Information Systems"

for copying the manual process too directly to the new medium.[32] Nevertheless, cartography and land use planning still constitute the predominant target application areas today and an understanding of their unique requirements and features is helpful.

Cartographic Systems

Automated cartography systems are primarily directed toward reducing the amount of manual labor required to produce conventional maps and charts. A collateral benefit is a reduction in the time delay between changes in the physical region and their reflection in the corresponding map products. In addition, data may be collected for only the lowest resolution level and aggregated to produce maps at a variety of scales. Typically, cartographic systems are implemented more in the form of a databank than of a full-powered database.

These systems often include a rich set of application-specific functions with which to manipulate, select, and depict features. At the same time, relationship information may be very limited. Many cartographic systems directly support only the "member of map" relationship for features stored in the database. In part, this is because the overwhelming majority of cartographic systems use the topological representation. That form does not easily support comparisons between geographic entities on the basis of location. It is entirely possible that the locations of two adjoining counties, for example, will be separately stored with the other information relating to those counties. Because each represents an independent entity, the system may very well not even recognize their proximity relationship. Such semantic information is left to the end user to extract visually from the

resultant maps.

Cartographic GIS applications generally deal with regions of relatively large extent. Statewide systems have been common for several years, and the same is becoming true today for systems of national and multi-national extent. Highly efficient methods of storage, manipulation, and retrieval are therefore essential. Well known cartographic systems include CGIS and KANDIDATS.

Land Use Planning

Geographic information systems to support land use applications stress manipulations involving descriptive and semantic data. Identification of spatial entities on the basis of their attribute values is very important in resource management, interpretation of demographic data, and urban planning. Data manipulations frequently involve relationships between named entities. Adjacency, distance, or containment criteria, for example, may be a basis for selection.

Many present land use systems are much like automated catalogs of geographic features, region characteristics, or surface attributes. Data is both collected and filed according to pre-defined regions. Such regions typically correspond to political or administrative regions, but geo-coordinate and arbitrary grids have also been used. Data retrieval usually requires the user to specify the pre-defined regions corresponding to the area of interest. Determination of the characteristics of arbitrary regions-- population and crop production figures, for example-- will almost certainly require special programming.

Maintaining the logical consistency of stored data is also a problem which is frequently solved by special programming. Each time a city's population is

updated, for example, a special program may be invoked to update the appropriate county and state records as well. Alternatively, population figures for the higher-level regions might not be explicitly stored, but instead repetitively computed each time they are required.

Land Information Systems (LISs) are an important subcategory of automated land use planning systems. A LIS maintains information on land characteristics which are relevant to legal actions, administration and economy, planning, and development. They are distinguished from conventional business-oriented DBMSs in that the data is related to real-world space. Information is stored and retrieved based on the location of political or cultural objects. The primary use of such systems is to retrieve maps interactively to display specified features and their surroundings. Frequently, the surroundings are of equal importance with the objects themselves. A typical LIS might, for example, contain information concerning a single town or district and include any of the following data:

1. Street names, postal-addresses of houses and their shape, position and use.
2. Boundaries, owner and use of land plots.
3. Position and attributes of pipes and electric lines.
4. Location coordinates of monuments, public buildings, and institutional structures.[13]

BASIS is a well-known urban land use planning system; WRIS is a good example of a system designed specifically for resource management.

4. A Location Data Set Approach

4.1. Rationale

The mechanism by which attributes, location, and entities are associated is a fundamental design issue. Attributes may be either attached to specific locations

or to specific spatial entities, or some combination of the two may be used. This choice will impact the types of operations which the system can perform.

The central idea of the Location Data Set (LDS) approach is that spatial data should be directly associated with locations rather than named regions or points. While this philosophy appears at first to run counter to conventional database design methods, on closer look it can be seen as an extension of the basic DBMS concepts of integration and controlled redundancy. Furthermore, it represents a more accurate model of the real world and thereby eliminates some artificial functional dependencies which complicate the database design problem.

The LDS approach recognizes that regions and points do not directly possess the characteristics of the locations over which they are defined. Rather they represent a *view* of the earth's characteristics over some subextent of its area, essentially a restriction on the global data. This is similar to the way in which the external views of a database represent a subsetting of the total data in the database.

There must be a distinction made between data associated with the view itself and data which exists totally independent of the view. Consider the case where a major city and its containing county merge to form a single political unit. The extent of the new city-county corresponds to a change in the view definition, whereas the population of the metropolitan area corresponds to base data and is unaffected. The population of the new unit represents an application of the updated view to the underlying population distribution. On the other hand, the name of the city manager and sales tax rates represent data which is truly functionally dependent on the geographic entity.

The LDS approach mirrors this reality by treating geographic entities as view definitions to be applied to various global data sets during data manipulations. These view definitions are referred to as *Location Predicates* (LPs). The term is intended to convey the meaning that the method by which a specific location, or set of locations, is specified should be irrelevant to the manner in which subsequent processing is performed. Thus, a location predicate may as easily correspond to a named region as to the results of a series of complicated selection operations relative to spatially distributed phenomena. The individual global data sets, one for each type of spatial data managed by a particular LDMS installation, are known as *Location Data Sets*.

Just as adding or eliminating view definitions does not affect an underlying database, so it is with geographic entities. In effect, each LDS is a single item database of global scope, against which the location predicates are applied. Any data representing phenomena that would continue to exist even if one or more defined spatial entities did not, is a candidate for representation as an LDS. Examples include population characteristics, vegetation types, mineral deposits, landforms, and a host of others. By storing such data in LDS form, the uncontrolled redundancy that would result from storing it with each separate geographic entity can be eliminated.

This perspective represents a degree of logical separation between geographic entities and their associated data which has been only occasionally hinted at in the past. Nagy and Wagle, for example, suggest that data organization may be improved if geometric and non-geometric data are separated, with the entity name serving as the link between the two. They point out that

> The distinction between geometric and value-related operations is a matter of the degree to which the respective attributes are used...it is useful to distinguish operations on data where there are no distinct entities and therefore no underlying geometry. In this case the geometry is *induced* by the values of a *surface variable* of the type $v=f(x,y)$ at every point in the area. When the values for v are *nominal,* this geometry takes the form of a partitioning of the area into regions of various types. [29]

Under this arrangement, the geometric data would be stored in a databank and the remaining data managed by a conventional DBMS. They do not extend the idea to the point of treating location data and the location of regions independently. Rather, they view region designations as simply a class of data values that may be assigned to grid cells or polygons.

The LDS approach adopts and extends this concept of cooperative management of data. Not only geometric data, but all data involving spatial relationships and distributions is removed to and managed by a separate GIS. Data which is truly functionally dependent on named geographic entities, such as the previously cited city manager example, would still be managed by the cooperating conventional DBMS. The idea of linkage through nominal identifiers is also extended by adding the notion of location predicates.

4.2. Design Strategy

LDMS stores a location predicate for every geographic entity explicitly declared to it; an entity's name and location predicate may be considered to be equivalent. Any location or set of locations may also be specified in terms of their properties, with the system calculating the corresponding location predicates. The user may, if desired, assign a name to the entity thus defined and subsequently reference it by that name.

The design of LDMS eliminates the need for special-purpose programming to maintain consistency across multiple resolution levels. This is accomplished by applying updates directly to their corresponding locations, rather than to any specific geographic entity. In this way, all entities which share the affected locations will inherit the updated information. This inheritance property applies even to geographic entities unknown to the system at the time of update but which might be de. ned at some subsequent point in time.

LDMS extends the concept of data independence for spatial data beyond that provided by most conventional database management systems. The definitions of data update entities and data retrieval entities are decoupled to provide *scale independence*. This means that the level of resolution at which data is inserted into the database is totally independent of the level at which it may be retrieved and manipulated. Indeed, a given installation may perform both update and query at multiple scales, ranging from the global level down to a resolution limited only by system resources.

4.3. Potential Advantages

The primary advantage of the LDS approach is its potential for enhanced data consistency. It is worth noting that the desire to eliminate redundancy, and its inevitable accompanying data inconsistencies, was a major impetus for the development of database technology. The problem is particularly severe when spatial data and shifting region boundaries are involved. Even in a well-organized, hierarchical region decomposition (e.g. nation, state, county, city) there should be consistency between corresponding data values at the various lev-

els. Where different types and multiple hierarchies of administrative or legal districts are involved, there may be additional problems caused by irregular decomposition, non-included regions, and areas corresponding to multiple intersections.

Another advantage of the LDS approach is its ability to quickly reflect drastic changes in region definitions while retaining access to the underlying data without conversion, update, or reprocessing. Consider the implications if, for example, boundaries of local governments undergo a major restructuring as they did in Great Britain in the mid 1970's. [35] In LDMS, all that need be done would be to add location definitions for the new administrative units, while still retaining the old ones. This would facilitate comparisons between districts of either type.

The capability to reference regions, features, and points by name, category, or characteristics-- as well as by coordinate location-- is often important. LDMS features in this area contribute greatly to the quality of the user interface. As an example of the flexibility inherent in the LDS approach, consider two alternative methods for determining the population of Dallas:

1. Direct query, such as "**LIST population OF dallas**". This method could be used if Dallas had been previously defined to LDMS.
2. Interactive identification. The user requests the system to "**SHOW texas**" or any other previously defined region known to include Dallas. Using the interactive graphics interface, the user indicates the city's location and queries the system for the population of the designated region.

5. Obstacles to Implementation

Any actual implementation of an LDMS prototype must overcome certain significant obstacles. Solutions to many of these are complicated by the LDS approach but are not unique to it. Those that fall into this category would need to be considered when designing any general-purpose GIS; they include global

extent, resolution, data volume, and access efficiency issues. Among those obstacles that follow more directly from the LDS approach, the identification of semantic data classes and the development of mechanisms for dealing with them stand out as perhaps the most challenging.

5.1. Global Extent

The *extent* of a GIS refers to the area which it covers. A general purpose system should be of global extent. That is, it should be capable of dealing with spatial entities located anywhere in the world. Furthermore, it must take into account that these entities may themselves extend over very large areas. Within the global reference framework, however, the system must also be able to focus on relatively small features: point phenomena of various types, cities, and small-scale legal or administrative divisions. This conflict is analogous to the problem of determining the proper balance between range and precision in numeric representation, and it complicates the design of a suitable reference strategy. This is in addition to user interface considerations of converting location coordinates between their internal logical form and that of the user.

While the use of spherical coordinates (latitude and longitude) might be preferable from an internal processing standpoint, that choice may complicate input/output operations. Conversely, a rectilinear grid reference yields cells of constant area and provides very simple location encoding. One option here is to place the origin outside the extent, so that all position encodings will be positive integers and distance calculations are simplified. If significant error is to be avoided, however, even large states must be subdivided and separate projections

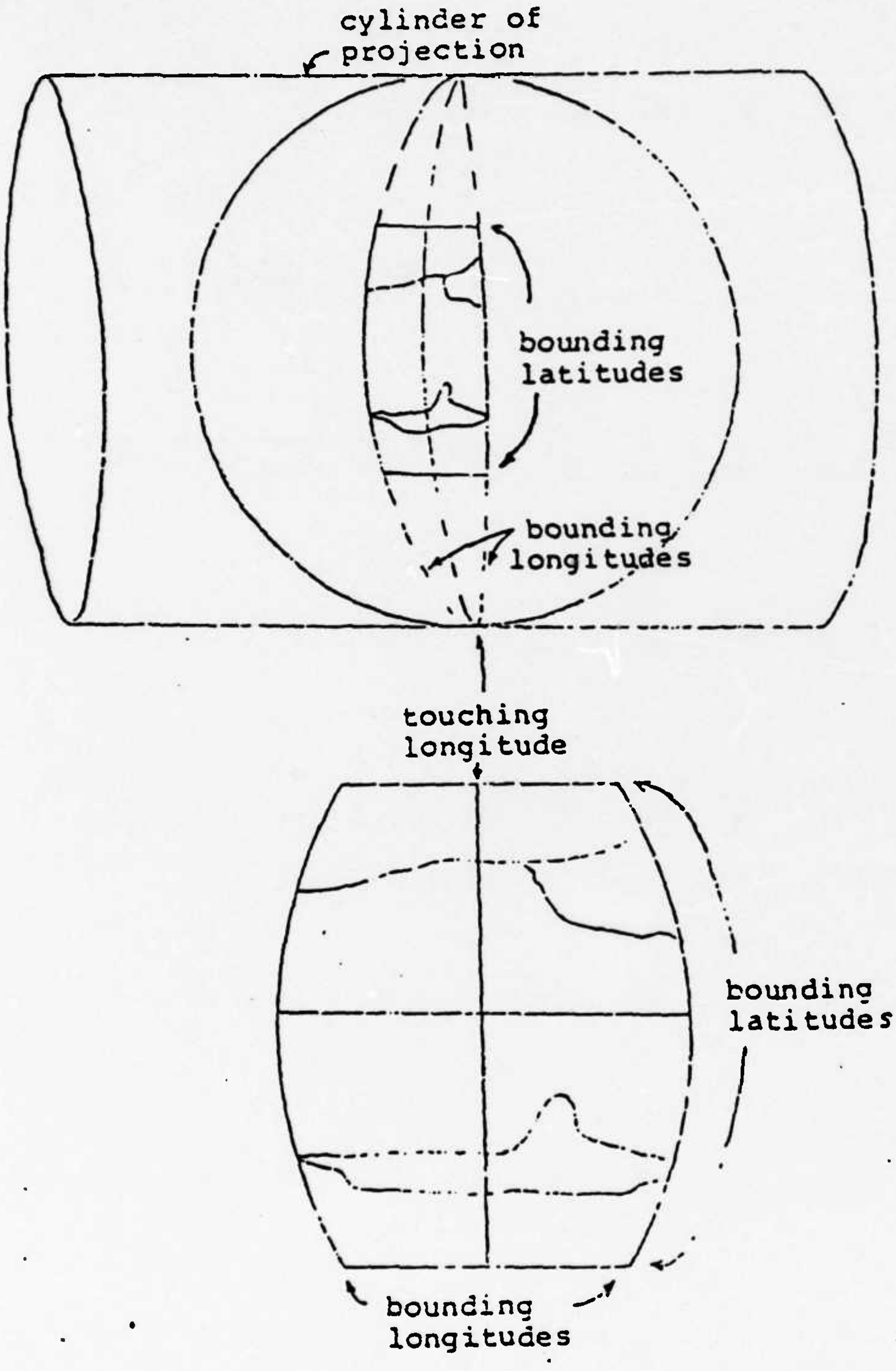
imposed on each resulting area.[29] The problem is compounded for a global system because it will likely combine data from a multitude of sources, gathered relative to an equally diverse selection of coordinate systems. This data will need to be converted, either during initial input or on an as-required basis during processing. Some of these conversions, such as those between range or township and their corresponding latitude and longitude, can be quite complex.

Another complication in systems of large extent is that the curvature of the earth must be considered when interpreting positional relationships and distances. The desire for compact internal location encoding formats may therefore conflict with the need to store information in a form amenable to such calculations. The earth's curvature complicates input and output operations as well. This curvature inevitably introduces distortions when its features are transferred to a flat surface, a process known as projection. In particular, it is geometrically impossible to simultaneously preserve relative distances, angles, and areas. Moreover, the significance of these errors increases with the size of the area depicted. Various types of projections obtain a higher degree of fidelity in some of these mutually incompatible properties, but only at the expense of increasing errors in others. Such considerations do not arise in systems of small extent which may safely ignore the curvature of the earth.

5.2. Resolution

A true GIS must provide for graphic display of output at various resolutions, which normally involves some degree of aggregation of the data from its stored form. This issue must be addressed from both the usage and time-space tradeoff

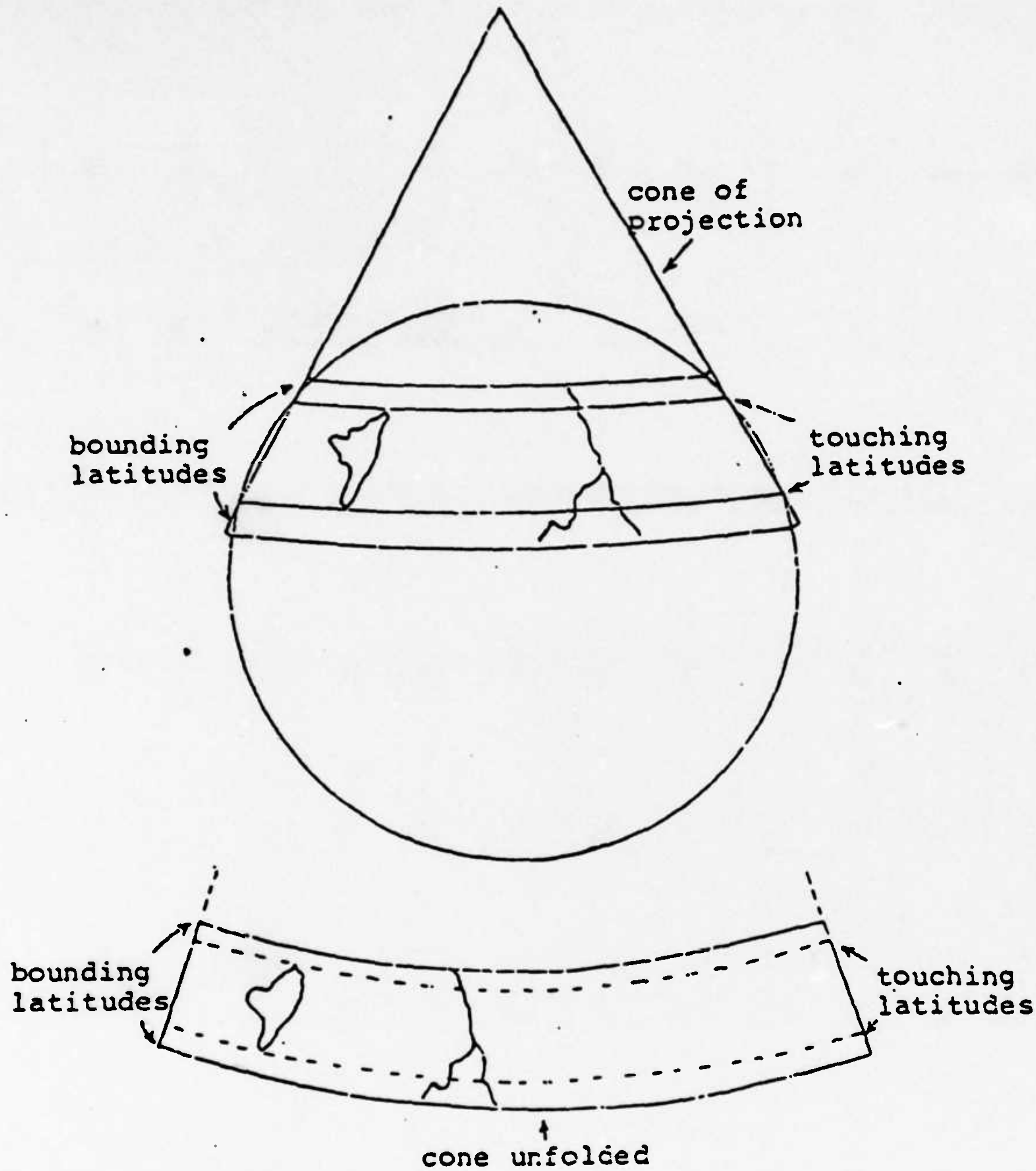
The Projection Dilemma: Angle Preservation



Transverse Mercator Planar Projection
To keep scale distortions small, the longitude coverage is restricted to a few degrees, and the bounding latitudes exclude the polar regions.

Source: Nagy & Wagle, "Geographic Data Processing"

The Projection Dilemma: Angle and Scale Preservation



Lambert Conformal Planar Projection
The scale is "true" along the touching latitudes. With proper selection of the touching latitudes within the desired latitude limits, the scale distortion at other latitudes can be minimized.

Source: Nagy & Wagle, "Geographic Data Processing"

perspectives. A multiple resolution capability also complicates the design of the user interface.

From the usage standpoint, the system must store data at the lowest level at which it may be potentially manipulated. While information at higher aggregate levels, say the population of the United States, can be derived from data stored at a higher resolution, say the population of each state, it is not possible to reverse the process. Higher resolution, however, usually comes at the cost of increased storage requirements.

In addition to the direct storage required to provide the required resolution, the processing time to derive aggregate data values must be considered.[22] This may involve the selection, storage, and placement of labels and features and therefore become quite complicated. Aspects of this problem include variable typesize, avoidance of over-writing significant features, and placement of multiple labels in subdivided regions.[29] Alternatively, processing time may be reduced by explicitly storing data at several different resolution levels.

When display scale and therefore resolution ranges over a wide range, establishing the user's context and reference neighborhood may become difficult. For example, the user may point to the same location at different times to indicate geographic entities of different types or scales.[12] Another problem that arises is that of distinguishing between multiple definitions of the same entity. This situation may occur when a previously-declared entity is entered again from other source data, or at a different resolution or precision. In practice, determining when two similarly-defined entities are in fact one and the same is usually carried out on the basis of a variance threshold test.[29] Selection of a suitable threshold

value becomes difficult when the geographic entities managed by the system range in size from point-source to regional and beyond.

Also in the area of user interface, the system must be capable of adjusting the output display scale over a wide range, and it should do so without requiring a lengthy or cumbersome dialog. A straight-forward solution is simply to expand the depiction of the requested region to fill the available display area. However, there is still the need to determine how much of the surrounding area should be included to provide a sufficient reference neighborhood. Furthermore, there is reason to believe that map images are more easily interpreted by viewers when they are displayed at standard scales, rather than the arbitrary ones produced by expanding the target region to the size of the screen.[12]

5.3. Data Volume

To put the data volume problem into perspective, consider that the land area of the earth is approximately 10**14 square meters. To represent the single variable of elevation at that resolution would consequently require 1.6 X 10**6 standard magnetic tapes. Although it is not yet technically possible to perform mass collection of data at this resolution level, that time is rapidly approaching. The Multispectral Scanner (MSS) of LANDSAT currently samples at 56 meter intervals, and the next generation of LANDSAT is expected to have resolutions of 30 square meters. [35]

5.4. Access Efficiency

Access efficiency encompasses several related issues. The types of data being managed, the stability of the database, and the required degree of support for

processing and display efficiency are all relevant considerations.

It is almost certain that the information density associated with some regions will be much greater than that of others. One would expect, for example, that more distinct types of data might be collected for land than for marine areas. It may also be the case that data collection constraints will result in more detailed or reliable data for some regions than others. The types of demographic data contained in the census reports of industrialized nations, for example, is simply not available for most third world countries. Furthermore, global extent implies the management of a potentially diverse range of data types, many of which will apply to or be collected for only limited regions.

A global system must be able to efficiently deal with this wide range of data density. It must allow sufficient capacity for high data density areas without pre-allocating, and thereby wasting, storage for similar data in areas to which it does not apply. This virtually requires some form of dynamic data definition and storage allocation scheme.

Support for processing efficiency must pay particular attention to data updates, as these are usually the most complex and error prone operations. Usually, processing costs can be reduced if the system is able to use problem-specific information and case occurrence statistics. Relevant information might include the spatial density of entities, number of entity types, data source and output product formats, volume of information, and geometric type of the entities.[29] Such information can also be used to cluster related data on secondary storage to improve access efficiency. Unfortunately, few assumptions can be made in these areas when the system is intended to be both global and general.

Because interactive graphics are an important part of the user interface, display efficiency considerations cannot be overlooked. The simple raster image approach suffers from serious storage inefficiencies. Therefore, a more commonly used technique is to represent each area by a list of polygon vertex coordinates, transform each coordinate to the desired scale, and then fill the refresh memory image values within the polygon under software control. This approach, however, is not optimum for interactive image displays. Digital image refresh memories are designed to drive TV rasters and are generally most efficiently accessed line-by-line. The polygon fill operation requires frequent random accesses to partial lines in order to build the display image. [7]

5.5. Semantic Constraints

A major complication is introduced by the diverse types of data which can be associated with location. This affects the manner in which data values are propagated from one resolution level to the next as well as the types of manipulation operations which may be appropriate. For these reasons, it is convenient to group data suitable for organization as location data sets into *semantic data classes*. That term is intended to convey the dependence of suitable data manipulation operations upon the nature of the real-world phenomena which that data represents.

The real world is infinitely more complex than even the most sophisticated models which might be developed to represent it. Therefore, it would be futile to attempt to enumerate all possible semantic data classes and their associated integrity constraints. Fortunately, there should be no practical need to do so. In

most fields of human endeavor, a small fraction of the total items represent the overwhelming preponderance of the workload, an observation formalized in the so-called 80/20 universal maxim. Therefore, it is reasonable to assume that a system capable of dealing with the most common semantic data classes will be able to handle most real-world data.

While this assumption reduces the scope of the problem considerably, it does not eliminate it entirely. Major semantic data classes must still be identified, along with their accompanying integrity constraints. The design of data structures to support these diverse semantic data classes and the operations appropriate to them represents a considerable challenge. There is an inherent conflict between the desire to devise data structures and logical operations common to all semantic data classes, and the need to enforce dissimilar integrity constraints.

The issue is further complicated by difficulties in determining the user's intent when alternative semantically valid interpretations are possible. When inserting the population value for a given city, for example, should the effect be to also increment a previously-entered state population figure, or should it be held constant? Similarly, what action is appropriate when population values have been entered for all counties in a state and the state population is later updated from a more recent data source? Should the previous county values be invalidated? Clearly, the answers to such questions depend on the nature and quality of the input data as well as the user's intent; all of these are, unfortunately, difficult for a machine to divine.

In addition to problems associated with the meaning of the data, there are similar problems involving the meaning of the geographic entities themselves. In

particular, two problems must be considered when managing time-series data: noncomparable geographic units and incompatibility of the cartographic and substantive databases.

The first situation frequently arises when the geographic units for which data were collected have changed. This problem of region redefinition is not limited to developing countries or infrequent changes in census units. A case in point: Although state and county boundaries have remained stable over a long period, the same is not true of smaller units. A U.S. Census Bureau survey of over five thousand incorporated areas showed that 56% of these underwent at least one boundary change in the 1970-74 time period.[24] Such changes can produce differences in data aggregation patterns which create the appearance of population shifts where in fact none exist.

The related problem of incompatibility occurs when valid data values are not available due to changes in the types of units, or a redefinition of the same units. This problem could arise, for example, when attempting to interpret historical data displayed over more modern base maps.

6. Existing Work With Application to Identified Obstacles

While the obstacles to implementation of LDMS are formidable, a considerable body of existing work can be applied to their solution. Much of this is specifically in the fields of GIS and DBMS design. However, the related areas of data structures, image processing, and computer graphics also make significant contributions.

6.1. Data Matrices

Two-dimensional arrays are the obvious choice for structuring data in grid-based systems. Their regular structure facilitates data manipulations involving the regions to which they correspond. Some of the variations on this theme have been termed data matrices, grid variables, and frames; distinctions between these are not always clear due to inconsistent use of the terms.

Matrices store the values of variables at a selected matrix of points in an area. One of the great advantages of matrix formats and their derivatives is the ease with which existing variables may be operated on to generate new derived variables. This is especially useful for overlay or composite analysis of two or more variables at a single location. Sophisticated overlay models may establish weighting factors for each of the participating variables. This method is commonly used to perform multivariate analysis of *surface variables*, especially the land use and land cover subclasses.[16] One might, for example, select all cells for locations corresponding to

```
(predominant_vegetation = conifer)
AND (elevation > 2000)
AND (elevation < 3000).
```

When the source data is collected over a range of different resolutions or is derived from irregularly-spaced observations, values for some grid cells must be estimated. This use of interpolation to compute discrete matrix values from continuous scalar-valued variables is called *gridding*. The two most popular methods of gridding are the *weighted average* method and the *trend surface* method.

The weighted average method sets the matrix value to the average for its immediate surroundings. This typically involves sampling those surroundings at a

higher resolution and weighting the value of each sample point according to its distance from the matrix point. The inverse square law serves as the default weighting option. [15]

The trend surface method involves two steps. First, a two-dimensional polynomial is computed to provide a least-squares fit to the sampled data points. This polynomial is then evaluated for each point in the grid to produce the values of the mapped variable at those points.

One variation of data matrices represents each region by multiple frames, one frame for each type of data maintained on that region.[4] Each individual frame may be thought of as one element in a stack of frames referencing the same region.

On the negative side, the multiple-variable grid cell structures used in virtually all grid-based systems (PICDMS is the only known exception) are often rigid. New types of data cannot be added for any region unless space is reserved for that purpose in advance. Not only does this waste storage until the fields are needed, but ultimate system expansion is limited by the number of preallocated blank fields. A further drawback is that new attributes must conform in type, length, and often name, to these predefined fields. [5] Matrix structures also have shortcomings when applied to the production of contour or *isopleth* maps. These arise because data values are stored for uniform increments of the *independent variables* but are displayed as uniform increments of the *dependent variable*. As mentioned previously, cell structures are also quite storage intensive. In selecting an appropriate grid resolution for output display, the general rule is that a grid size one half that of the smallest feature is required to retain the detail of the

map.[16]

6.2. B-Trees

Most of the tree-like data structures implemented or proposed for geographical information systems are derived from B-trees, themselves a generalization of binary search trees.[6] The attraction of B-trees is that they remain balanced while acquiring and releasing storage dynamically. Storage efficiency of at least 50 percent is guaranteed, and this may be increased substantially through certain optimization measures.

The cost of a random access in a B-tree is proportional to the height of the tree, itself a function of the fanout factor. Therefore, access speed can be improved by packing more entries per node. Lomet has proposed a structure, called digital B-trees, which provides some of the speed advantages of hashed access with the sequential ordering of B-trees.[23] The method involves doubling node size (through the use of multiple-page nodes) as an alternative to splitting. Records are assigned to pages in the nodes in an ordered fashion based on the binary digits in their keys. This distribution strategy maintains the ordering of records within nodes, while allowing an immediate determination as to the page of a node on which a given record resides.

Digital B-Trees appear to be a good choice for organizing records that represent spatial objects. One technique that has been used to create key values for such records involves interleaving of the binary digits corresponding to the object's coordinates.

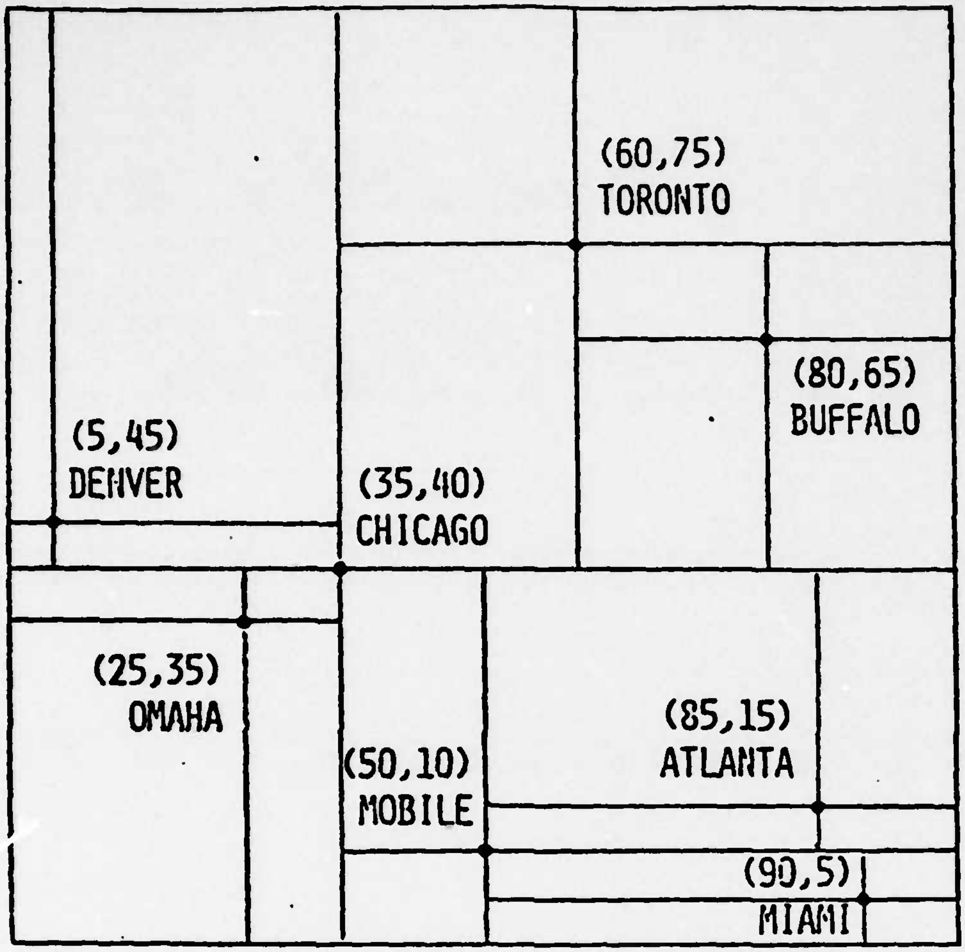
6.3. Quadtrees

Quadtrees are a multidimensional generalization of binary search trees, originally proposed by Klinger[18] and subsequently developed by various other researchers, notably Samet.[36,37,38,41,25,39] They are a class of hierarchical data structures based on the principle of recursive decomposition of space, typically through the partition of a region into a set of maximal blocks. As is true of tree structures generally, quadtrees are particularly well suited to performing search operations efficiently. Traditional region representations such as the boundary code are very local in application, and make it difficult to avoid fairly exhaustive searches for region characteristics. Quadtrees, on the other hand, are more global in nature and enable the elimination of large areas from consideration.

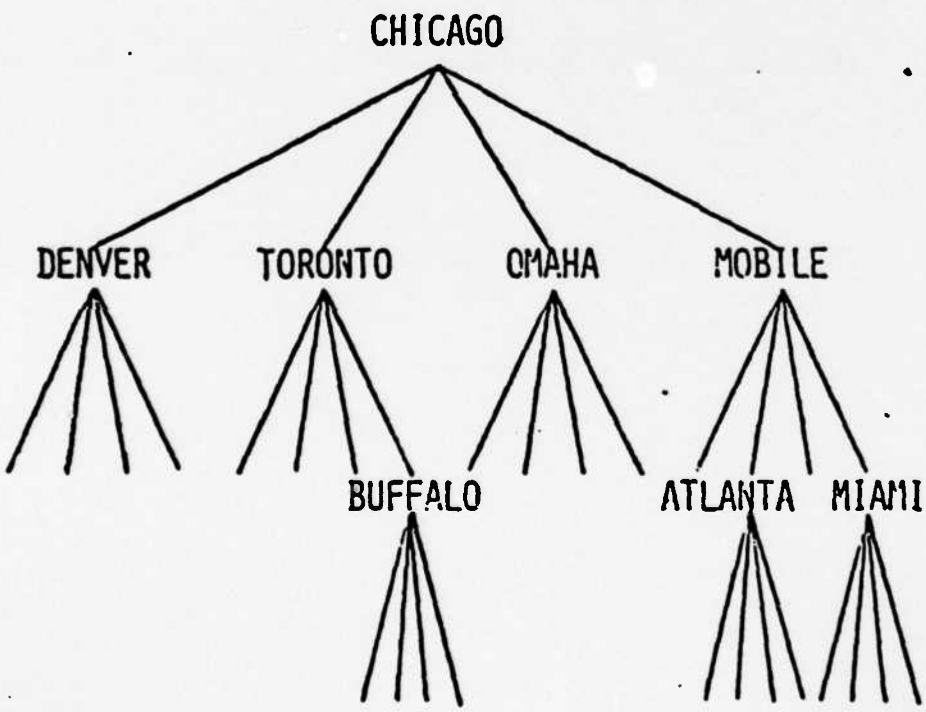
Most algorithms suitable for operations on pixel or grid data apply to quadtrees as well, and its more compact form usually permits faster execution. Set operations such as union, overlay, and intersection fall in this category. The hierarchical structure is well suited to identifying containment relationships. Quadtree algorithms often require time proportional to the number of blocks represented, independent of block size.[36] Quadtrees can be differentiated on the following bases:

1. The type of *data* that they are used to represent
2. The principle guiding the *decomposition* process
3. The type of *resolution* (variable or fixed) [39]

Seemingly endless variations on the quadtree theme are possible. They have been used to represent regions, curves, volumes, and collections of point data. Depending on the implementation and the nature of the data referenced by a given tree, that data may be:



Example of partition into non-equal size quadrants



A quad tree and the nodes it represents.
Source: Samet, "Hierarchical Data Structures for Representing Geographical Information"

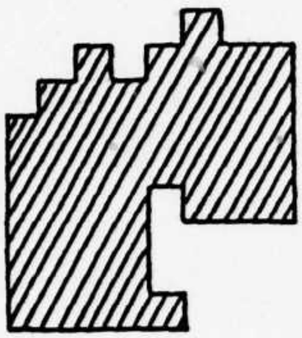
1. Stored only in leaf nodes. This approach is useful for decomposing image data into partitions while minimizing storage costs.
2. Stored at the lowest resolution in leaf nodes, with aggregate values propagated upward and stored in ancestor nodes. This method permits fast pruning of subtrees during searches. It also allows retrieval of approximate values or reduced-resolution images to be performed at lower cost.
3. Stored in either an internal or a leaf node. Records representing spatial objects can be stored at the most appropriate level, perhaps based on the object's size or set membership type.

In true quadtrees, each data node in the tree possesses four subtrees, commonly referred to as quadrants or subquadrants. Due to their two-dimensional nature, these provide an efficient structure to reference spatial entities defined in two dimensional coordinate systems (such as lat-long). Generalizations to more than two dimensions also exist. Region decomposition need not necessarily be into equal-size quadrants. Rather, regions and storage can be allocated based on the *quantity of data associated* with each quadrant. A variant known as k-d trees uses this method.

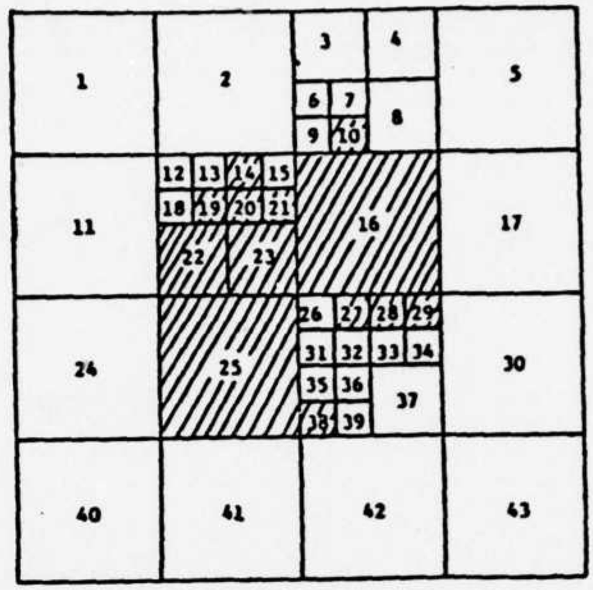
K-D trees are reported to provide improved efficiency in both storage and retrieval, due in part to better balance of the tree. Maintenance of the tree is somewhat more complicated, however. As a practical matter, some of the potential improvement in storage efficiency must be sacrificed to simplify deletions from the tree.[25]

There are two major approaches to region representation and versions of quadtrees exist for both of them: those that specify the boundaries of the region and those which organize its interior. A basic *region quadtree* successively subdivides an image array into four equal-sized quadrants. The method presumes some ultimate, fixed final resolution, perhaps at the individual pixel level. The trees are

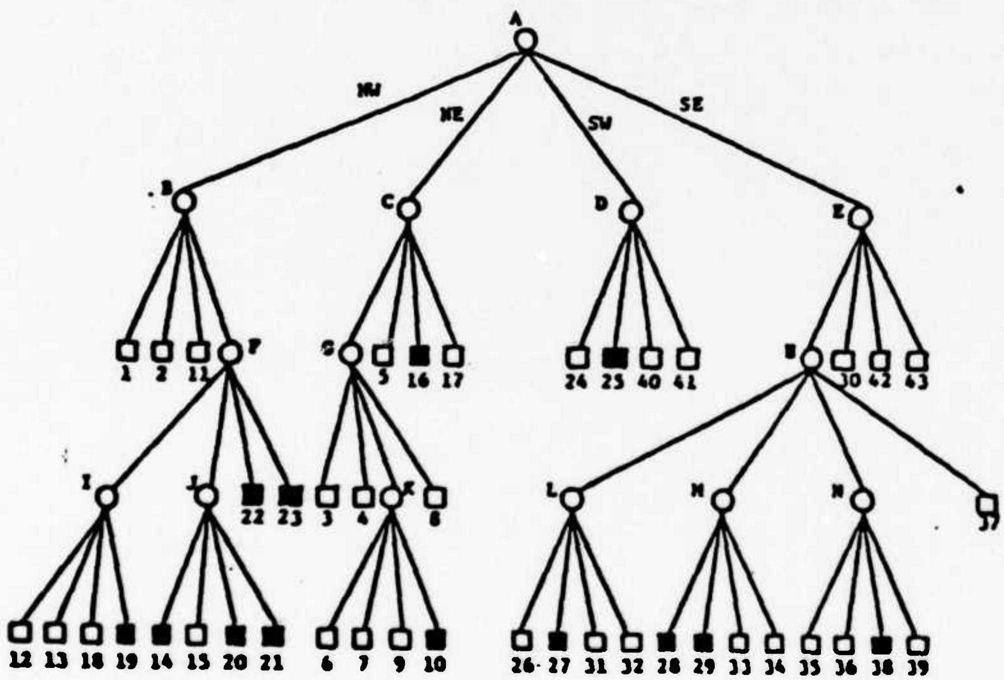
Quadtree Representation Involving Region Decomposition
Into Equal-Size Quadrants



a, Region



b. Block decomposition of the region in (a).



c. Quadtree representation of the blocks in (b).

A region, its maximal blocks, and the corresponding quadtree. Blocks in the region are shaded. Background blocks are blank.

Source: Samet, "Hierarchical Data Structures for Representing Geographical Information"

composed of leaf and nonleaf nodes. For a binary image, leaf nodes may be either BLACK or WHITE, whereas all nonleaf nodes are considered to be GREY. There are also gray-scale variations in which each leaf node may assume one of a range of values. In these, the values of nonleaf nodes are a generalization of the values contained by their descendents. [39]

One of the problems in implementing quadtrees as actual tree structures is the amount of storage and processing overhead that form entails. An image consisting of *B* BLACK nodes and *W* WHITE nodes, for example, requires storage for $(B + W - 1)/3$ internal GRAY nodes. Furthermore, each node requires room for pointers to its sons. These costs have been reduced somewhat in various pointerless implementations. [39]

6.4. Pyramids

Pyramids have their origins in the field of image processing. They are a close relative of region quadtrees but differ primarily in that their resolution is fixed. This produces an exponentially tapering stack of arrays, each of them one quarter of the size of the previous array. In a sense, they are quadtrees that are both regular and complete, and the two structures share many desirable properties.[39]

Data stored in all except the highest resolution array is essentially redundant; it could be computed when required by aggregating data at a lower level. Storing it explicitly, however, may eliminate a considerable amount of repetitive processing during data manipulation. This makes pyramids especially useful for applications requiring a rapid zoom capability for displayed images.[26]

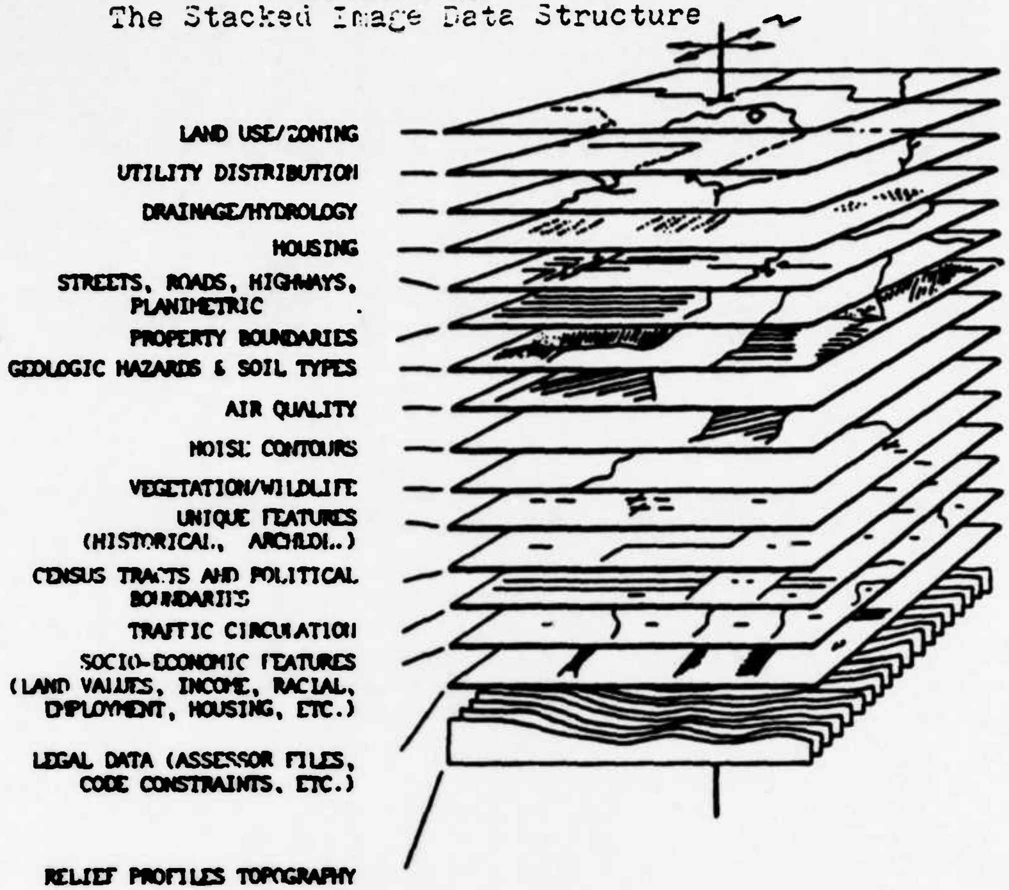
6.5. Dynamic Stacks

Frames are useful for associating attribute values and locations without the need to explicitly encode position coordinates with each grid cell value. Instead, the position of an entire frame of cells is registered; the position of each subarea can then be derived from the regular matrix structure. For large databases, use of frames may still be a problem unless some sort of compression scheme is used to reduce storage requirements.

While several grid systems have adopted the idea of stacking multiple frames, PICDMS is unique in that it treats the stack as a dynamic structure. Not only does this allow new base or derived data categories to be added without a major restructuring of the database, but it also reduces the wasteful pre-allocation of storage to regions where a particular type of data may not apply.

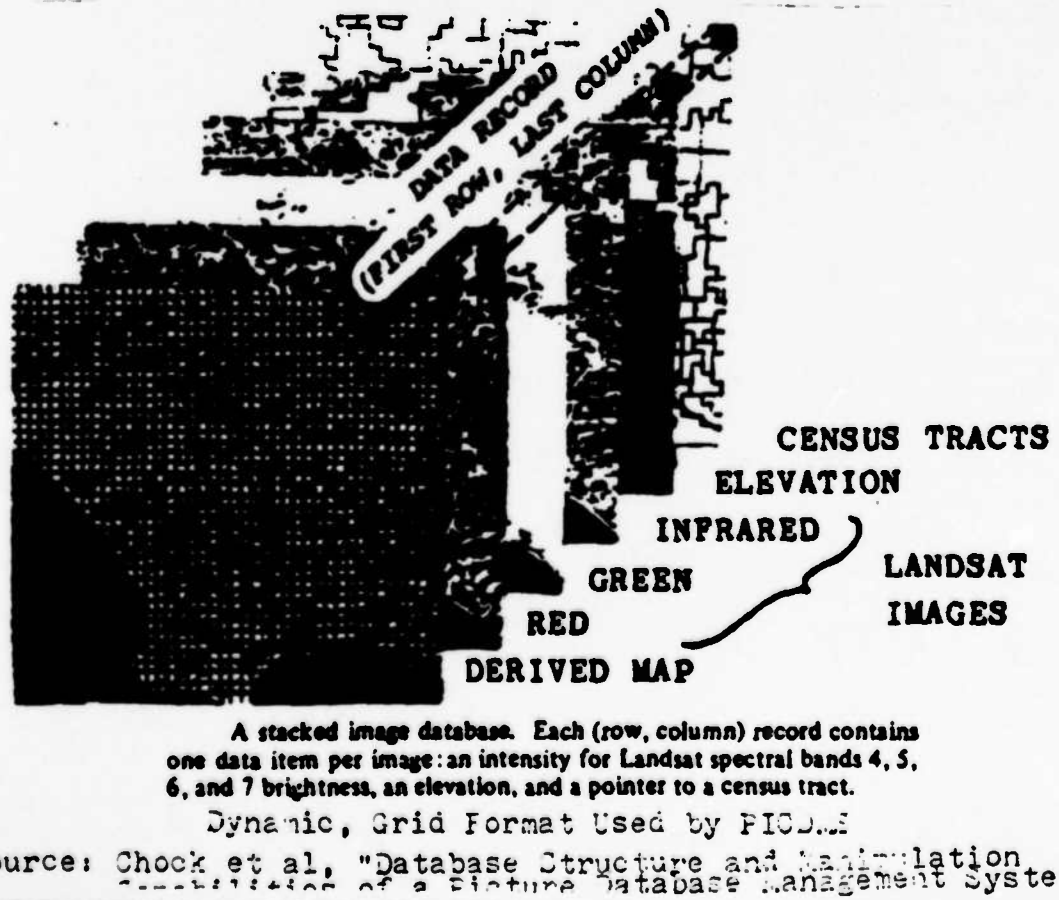
While the dynamic stack method of PICDMS offers both expandability and flexibility, storage costs may still be quite high where large regions are involved. There is no provision for compression of redundant data values, although extensions to deal with that problem have been proposed. [21] Furthermore, some of these values may themselves be quite storage intensive. One example would be landmark labels, such as "Mississippi River" or "Dodger Stadium", repeated for possibly thousands of grid cells corresponding to the landmark's location. To make matters worse, storage is allocated for each cell of the entire frame, even if most of these are not destined to hold labels. [5] Thus, dynamic stacked frames eliminate the waste caused by preallocating storage for the entire system extent (i.e., all regions), but not that produced by preallocntion to entire regions.

Variations On
The Stacked Image Data Structure



Multi-Layer Information System (Foster, 1977)

Pre-Allocated, Topological Format



A stacked image database. Each (row, column) record contains one data item per image: an intensity for Landsat spectral bands 4, 5, 6, and 7 brightness, an elevation, and a pointer to a census tract.

Dynamic, Grid Format Used by PICDMS

Source: Chock et al, "Database Structure and Manipulation Capabilities of a Picture Database Management System"

6.6. Compression Techniques

Some form of constant compression for null or redundant values is essential for large data bases based on a grid or tabular structure. Without it, the conflicting requirements for reasonable data resolution and storage costs cannot be met. Both location predicates and location data sets are candidates for the use of such techniques. This will inevitably involve some additional processing overhead to perform encoding and decoding of compact representations.

A number of compression techniques have been implemented or proposed for B-trees which also have potential for more general application. Elimination of common prefix information in a series of clustered values is one example. Encoding, by use of a lookup table or other means, is an option. Pointer compression is another conventional DBMS technique which is especially useful in virtual memory systems with large address values. In some cases, explicit pointers may be eliminated altogether; Samet describes a method to encode and compress image data by storing it as a pointerless quadtree.[38] Other methods of reducing the storage requirements of quadtree-based structures involve encoding and compression of the leaf data. Data at the lowest resolution level may be stored as compact topological codes, or the point sets may be compressed. Hypercube encoding[2] is one of several methods proposed to generate compact encodings of point sets.

Eggers and Shoshani have proposed a compression technique which allows a high degree of compression but requires only logarithmic access time.[10] Their method employs a constant suppression scheme which may be iteratively applied to produce single encodings which represent a range of data values. It is

especially suited for encoding vector (matrix or array) *data in stable databases.* The conversion scheme encodes the location of constants and suppressed values into a corresponding, but smaller, header vector of counts. Sparse matrices are a particularly good candidate for suppression with this method. The compression factor is greatest for vectors in which the compressed values are highly clustered, as might be expected for geographic applications. Even in the worst case, however, the method will produce header vectors of the same size as the uncompressed vector. Because of this characteristic, the file header method is a good choice for implementing file transposition of a database. This strategy involves partition of a base attribute file into a set of derived files according to the attribute values. Each of the derived files can then be encoded. The resulting benefits are two-fold: storage costs are reduced and objects related by common attribute values can be more efficiently identified.

Bit map and run-length encoding are alternative methods to achieve storage compression through constant suppression. The bit map method involves setting the bits of an array, or *bit matrix,* to indicate which vector positions are being suppressed; run-length encoding uses value-count pairs to encode constant values appearing in series. Both of these generate a greater overhead in access time than does the header vector method. While the bit map and header vector methods produce an equal compression of the source data, the bit map headers may be smaller if the average length of a constant series is greater than sixteen.[10]

One method in common use for image data is to compress raster structures in one direction. The resulting set of scan lines retains the ordering of the data but no longer supports equally efficient traversal in both the *X* and *Y* directions.[33]

Another technique is known as *autoadaptive block coding*. It uses the binary digits to compress a hierarchical image representation. In that scheme, *0* represents a block composed solely of WHITE pixels; *1* represents a GRAY or BLACK block to be recursively divided into four subblocks.[39]

Vector (topological) representations are much more storage efficient than are equivalent bit matrix representations at the same resolution. Therefore, one possible form of data compression is to incorporate topological primitives into an underlying grid structure. The contributions of Guzman, who proposed an adaptive data representation approach based on regular decomposition of a region, are relevant here. His design provided for data to be stored in the leaves of a hierarchical structure, with the format of the stored data dependent on the types and frequencies of operations performed on it. There were also provisions for automatic conversion of the data between vector and grid formats as usage patterns changed.[17]

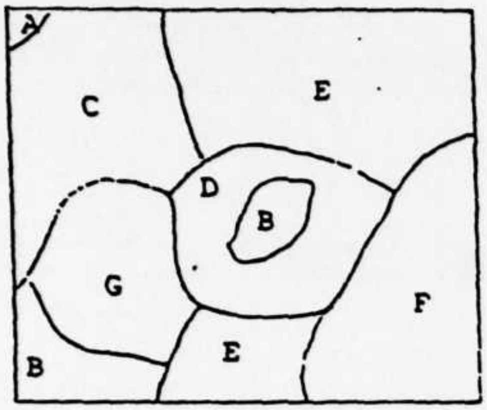
6.7. Format Conversion

Each of the two data formats, topological and grid, is able to represent some types of spatial objects more compactly than other types. Similarly, each is more suited to some types of data manipulations than others. The choice of one over the other can therefore have a major impact on both storage and processing efficiency. The form of the source data is certain to be a major consideration in that choice. Remotely-sensed and scanner-digitized data, for example, represent an increasing proportion of total input data and these are received in raster grid form. Unfortunately, the topological format is usually a better choice from a pro-

Data Format Selection and Conversion Dilemma

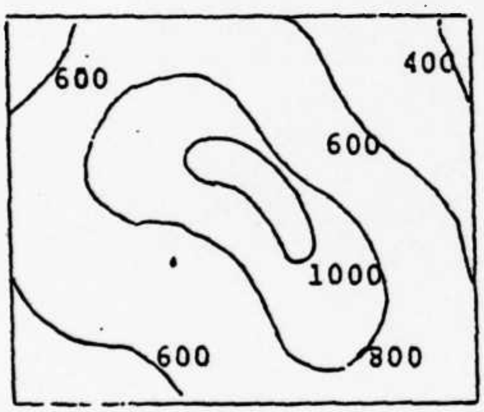
POLYGONAL REPRESENTATION

CELLULAR REPRESENTATION



C	C	E	E	E	E
C	C	E	E	E	E
C	G	D	B	D	F
G	G	D	D	F	F
B	B	E	E	F	F

a. Nominal Variable (Thematic Map)



400	600	600	600	400	400
600	800	800	600	600	600
600	800	800	1000	600	600
600	600	600	800	800	600
400	400	600	600	600	600

b. Scalar Variable (Contour Map)

- <u>Dual Representation of Surface Variables</u>

  The cellular representation is obtained from the polygonal representation by determining the dominant value of the variable in each cell.

  Note that information is lost in the conversion from topological to grid representation and the process is therefore not reversible.

Source: Nagy & Wagle, "Geographic Data Processing"

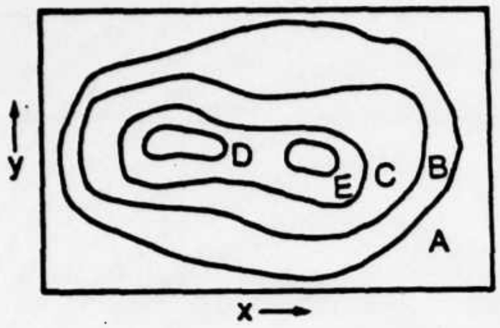
cessing efficiency standpoint.

Especially in cartographic systems, it is frequently less expensive overall to convert data stored in grid format to topological form for processing, and then back again to raster form for output display. These operations form an important group of processing functions and may represent a significant computational overhead. [33] Other options are possible and, in general, cartographic GIS designers have responded to the data format and conversion dilemmas in one of three ways:

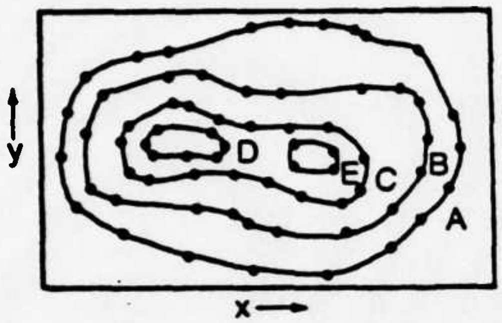
1. The most common approach is to store data in a raster grid format similar or identical to that in which it is received. Conversion to topological format is performed as a preliminary step to analytical or manipulative processing whenever it is advantageous.
2. A second option is to convert at least part of the input data to topological format before storing it in the database. This choice is attractive when no raster-mode counterpart exists for a known vector-mode algorithm, or if most manipulations upon a given class of data are vector oriented. IBIS takes this multiple format approach and provides operations for use with image, graphics, and tabular data types.[45]
3. A third possibility is to store data in a hybrid data structure which possesses characteristics of both raster and vector forms. One such structure, dubbed VASTER, could be efficiently manipulated using algorithms based on either of the two primary types without the need for intermediate conversion.[33] *Raster-encoded polygons* are another hybrid form. These use a modification of the run-length encoding scheme, widely used for storage compression of data in raster format. The technique involves the addition of three supplemental fields, specifying distances to the nearest state and county boundaries, to the encoded definition of each raster line. [7] One of the obvious limitations of raster-encoded polygons is their lack of generality; the physical data structure format is inextricably tied to the logical political hierarchy being represented.

Polygon to grid conversion is relatively straight forward, and is routinely handled by scanners and other mass digitization devices. Grid to polygon conversion is not as simple, but has been demonstrated by systems such as POLYGRID.[20] A GIS used by the Data Analysis Laboratory of the U.S. Geological Survey possesses a similar capability. [15]

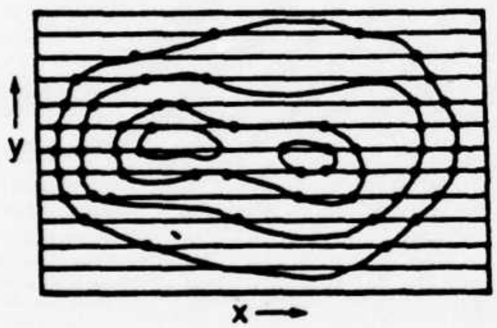
One Possible Solution to the Data Format
Problem: Hybrid Types



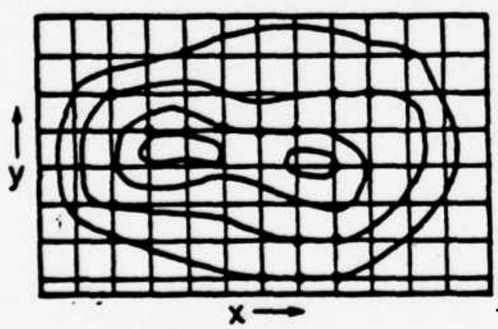
Original Contour Map
(Analog)
(Traditional Map)



Vector Organization
(logical record =
contour line)
(Topological)



Raster Organization
(logical record =
horizontal strip)
(Note: Often Considered to be
a Subtype of the Grid Format)



Grid Organization
(logical record =
grid cell or vertex)

Types of data organization

The Proposed Hybrid Type, VASTER, Would Combine
Scan Lines from the Vector Format with Chain
Encoding of the Area Between Scan Lines

Source: Peuquet, "Vector/Raster Options for Digital
Cartographic Data"

7. Survey of Existing Systems

Several existing systems possess specific features or design elements which could be adapted or extended for use by LDMS. This section will briefly survey a number of such systems, noting design strategies which could be incorporated into an actual LDMS implementation. The systems chosen are not necessarily the best or most representative examples of GISs in general, and only those aspects most relevant to the design of LDMS will be addressed. The reader desiring a more comprehensive survey of the state of GIS design is referred to the excellent tutorial by Nagy and Wagle.[29]

7.1. BASIS (Bay Area Spatial Information System)

BASIS is a grid-based urban planning system which fixes the number of cells, the maximum resolution, and the cell record format. It has the capacity to store 80 data items per cell at a resolution of one hectare. Its designers selected the grid cell structure because the cell coordinates double as an implicit storage structure. The location of specific hectare data cells on disk is directly computed as a function of that cell's location within a one-kilometer square.

Base maps are input using digitizer tables. It is also possible to later change the resulting values of specific map cells using interactive terminal-based editing programs. [44] This feature is useful for correcting errors introduced during mass data input operations, or to update data for a limited area.

7.2. BROWSE

BROWSE is an interactive raster image display facility developed at

Carnegie-Mellon University as the front-end to an integrated Map Assisted Photo-interpretation System (MAPS). [26] A multi-resolution image database contains aerial photograph images of a region, and these are organized hierarchically according to their resolutions. The resulting pyramid structure allows a rapid zoom capability at the cost of redundantly storing the same image at several resolutions. The BROWSE package is partitioned into four software levels corresponding to the user interface, supporting subroutines, window management, and graphics primitives. An interesting feature is the creation of temporary files by the window package routines for subsequent processing and display regeneration.

7.3. GADS (Geo-Data Analysis and Display System)

Developed by IBM, GADS is a sophisticated and flexible GIS. It is aimed at supporting unstructured interactive problem solving for urban applications. The user interface is strongly oriented toward interactive graphics operations.

GADS uses geographical regions, called *zones*, as objects; it stores their non-spatial attributes in a DBMS and their boundary coordinates in a separate system. [5] Zones may be defined either as strictly geometric entities such as uniform squares, or they may correspond to units natural to the application, such as city blocks or school districts.[29]

GADS supports the concept of overlay maps, or *superzones*. These are aggregations of previously-defined basic zones. Like the location predicates of LDMS, their definitions may be stored for later use.[31]

7.4. KNRIS (Kentucky Natural Resources Information System)

A 15-minute topographic map series serves as the basic unit of data organization for KNRIS.[1] These are assembled by composing mosaics of 7 1/2-minute topographic quadrangles. The 15-minute map modules serve three purposes:

1. Map and data organization.
2. As a stable base for digitization.
3. As an inexpensive reproduction of interim map products.

KNRIS data is organized into files known as *manuscripts* on the basis of polygon regions termed *integrated terrain units (ITU)'s*. Physical data is represented as points, lines, and polygons. Resolution for the minimum size land use unit is set at two acres.

A major design consideration was the desire to develop interfaces between KNRIS and other resource management information systems. There has been particular success in exchanging information with the Area Design and Planning Tool (ADAPT) system, a commercial product. ADAPT uses a triangular irregular network (TIN) data structure. This structure was developed as a data-entry device for LANDSAT data and is based on a cylindrical projection. Terrain data, for example, is represented as a geographic coordinate and elevation pair associated with each vertex. [1]

This capability to utilize existing data files without expensive reprocessing is highly desirable. In particular, LANDSAT, USGS, and census data should be directly usable by any GIS which is intended for users with minimal data collection resources at their disposal.

7.5. LIS (Land Information System)

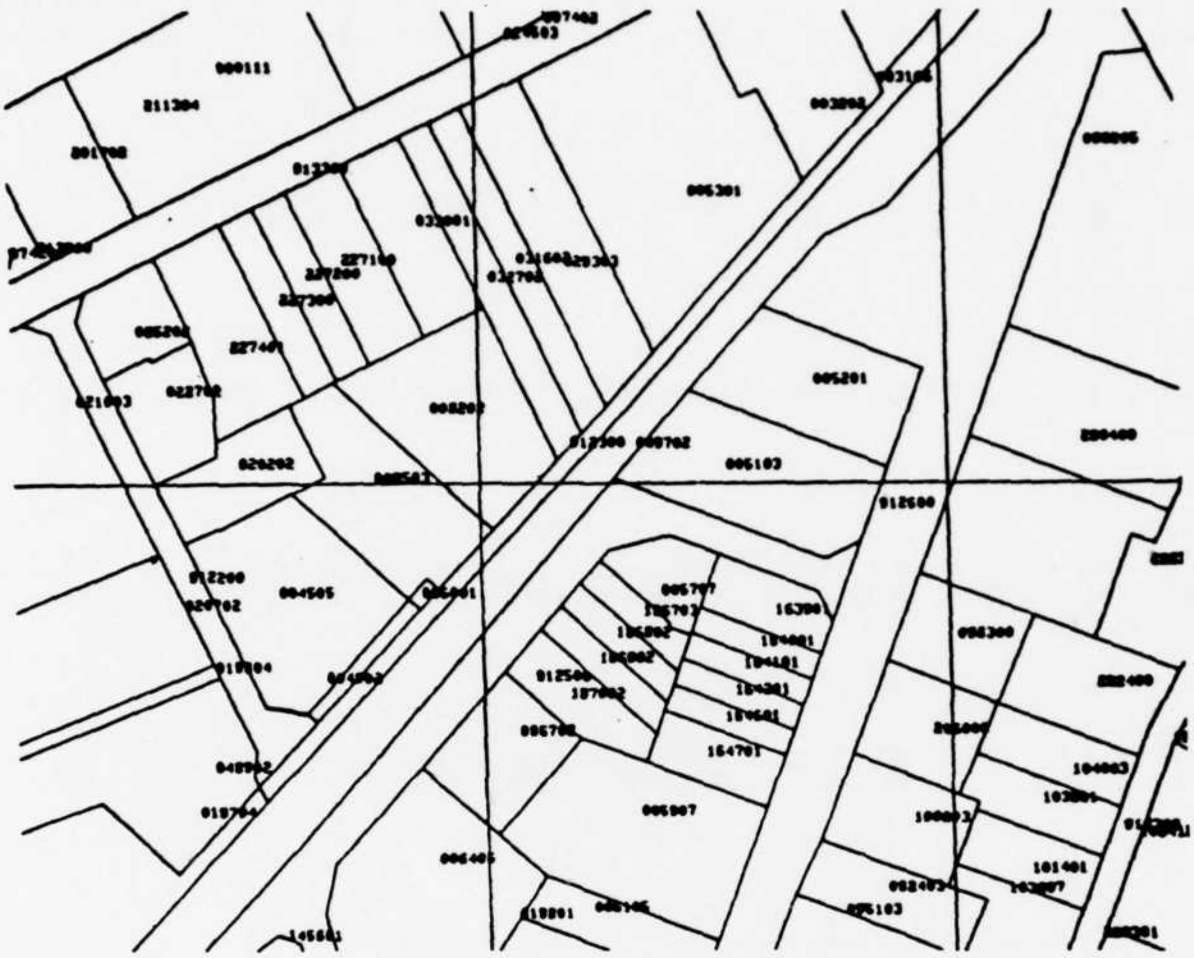
A Land Information System developed at the *Zentrum fur Interaktives Rechnen* in Switzerland was implemented using an adaptation of a CODASYL network model DBMS.[13]

The underlying operating system provided the means to specify the physical placement of stored records on the disks. This capability was an essential requirement, as the design relies heavily on the clustering of stored data according to its geographical neighborhood. The conventional file system and access methods were bypassed and more suitable methods substituted.

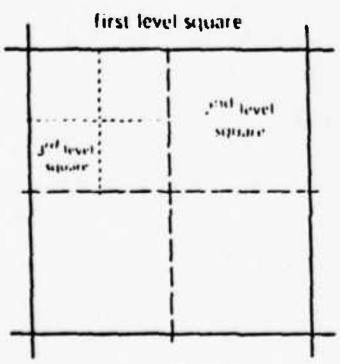
The basic solution adopted was to impose a uniform reference grid over the land area and to assign a unique number to each square. The address space was divided into corresponding pages and each grid square mapped to a page. A record representing each real-world object is stored on the page associated with its grid square. The actual image representations of objects are stored in topological form.

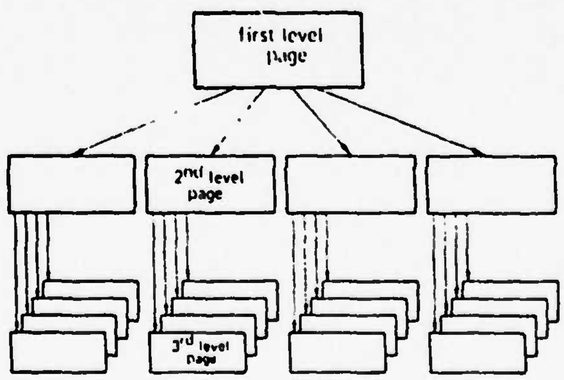
In order to enhance processing efficiency, several refinements were incorporated into the basic design. When the number of objects referenced by a particular grid square exceeds the capacity of a single page, the grid is subdivided into four smaller squares and each quadrant is mapped to its own physical page. This subdividing may be carried to an arbitrary number of levels. Each level maintains pointers to its sub-squares, so the method represents a variation of the quadtree data structure in which records are stored in both internal and leaf nodes.

Hierarchical Access Structure for a
Land Information System



Top Level Division of the Region for First Level Pages





Source: Frank, "Application of DBMS to Land Information Systems"

Assignment of Records to Pages

Figure A.

Object Records Are Assigned to the Page Which Corresponds to the Smallest Square Wholly Containing the Object.

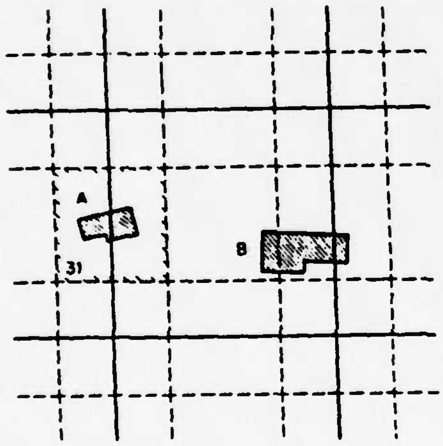


Figure B.

Offset of Squares at Next Lower Level.

Assignment of Records to Pages/Squares May be Further Restricted Due to Maximal/Minimal Size Constraints.
(Neighborhood Factor)

Source: Frank, "Application of DBMS to Land Information Systems"

Object records are always stored on the page corresponding to the smallest grid section on which that object will fit completely. To avoid propagation of higher-level grid lines through all subordinate levels, subdivision lines are offset at each level. This ensures that those objects that straddle grid boundaries at one level can be placed at the next level. Display of image data is straight forward; object records are retrieved and the image materialized by interpreting the topological descriptions which they contain.

7.6. LUIS (Land Use Information System)

LUIS is a generalized system suitable for handling any database of geographically-oriented integer information. Its designers recognized that all geographic information has two elements: the *location* and *description*. Therefore, a system to handle this type of information must relate these two components. LUIS does this by separating the data into *plot files* and *database files* and linking the two through physical pointers. Plot files are further divided into pointer file and coordinate file components. Pointer files contain fixed length records consisting of polygon centroids and bounding rectangles, in addition to pointers into the coordinate and plot files. The coordinate file component contains sets of point vectors which define the polygon perimeters more precisely. Through the use of *keys*, or physical pointers, to link entries in the location and database files, polygon locations and their other attribute data are actually handled together as single logical units.[28]

Bounding rectangle information is used to simplify logical operations and quickly determine the appropriate scale for output display. New regions (called

maps) may be defined through operations on previously defined regions and used in subsequent display commands. Also, multiple regions may be superimposed or otherwise jointly displayed.

A subset of the LUIS map plotting commands could be implemented in LDMS. These include:

DRAW mapname
Draw the named map on the CRT at the largest possible scale. Maps may be either standard system maps or subsets of these previously defined by the user.

DRAW mapname USING dataname
Draws only those polygons in **mapname** associated with values of **dataname**.

DRAW mapname SHADED USING dataname
Draws the polygons in **mapname** and shades them according to the values of the variable associated with them.

BLOWUP
Redraws a user-designated section of the currently displayed map. The section is expanded to fill the entire available screen area.

SUPERIMPOSE mapname
Exactly the same as **DRAW** except that the screen is not cleared, but rather the new map is overlaid at the scale of the current display.

DEFINE newmapname type
Allows the user to define an area that can later be used in a **DRAW** command or used as an input to a further **DEFINE** operation.[28]

Applications envisioned for LUIS include plotting the location of medical or other service facilities; studying the location of diseases in epidemiology; and analyzing housing development, demographic trends, and crime patterns.[28]

7.7. MAPSOFT

The MAPSOFT system at Akron University uses a vector representation to produce files defining the outlines of polygon regions or collections of point locations. These are known as *locational data files*. Each locational data file is paired with a corresponding statistical data description file containing attribute data asso-

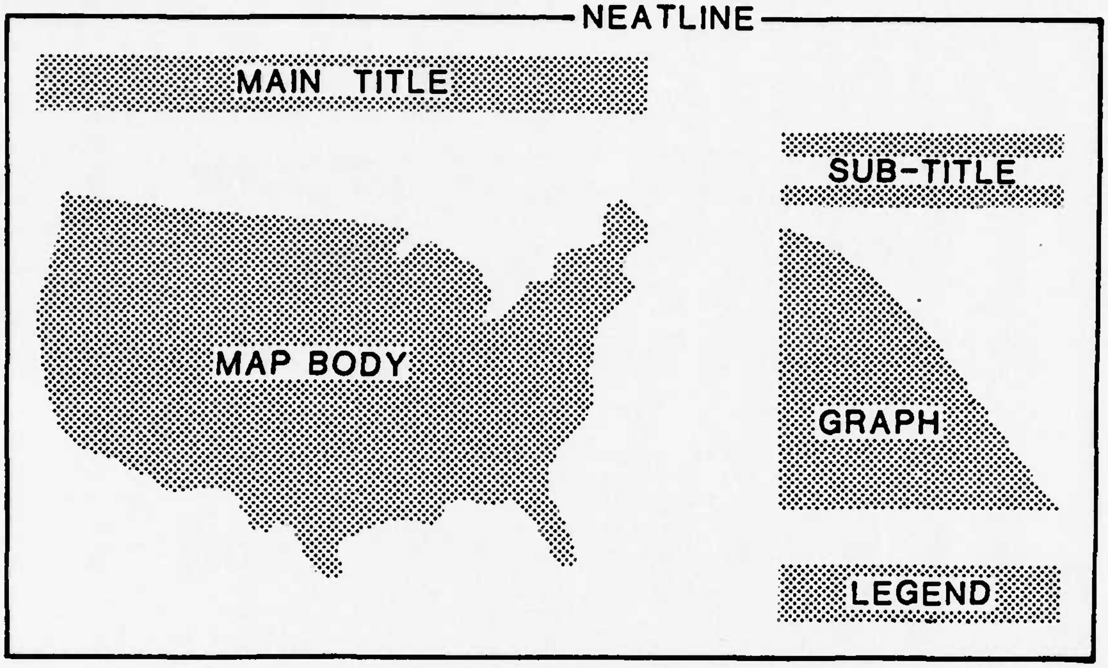
ciated with its regions or points. Both locational and statistical data files are ordered sequentially, with data for a given entity entered in the same ordinal position in each of the paired files.

The generation of a statistical map requires a series of two-step operations. Each operation involves the extraction of locational information from one file and the processing of data in its corresponding statistical data file. The basic sequence consists of plotting region outlines, plotting point data, and displaying the statistical characteristics of the data. These characteristics may be depicted by a variety of choropleth (cross-hatching or shading), cumulative frequency distribution graphs, or bar graph variations. [43] Each named point or region is associated with a specific set of attribute values, fixed in number and collectively linked to one and only one entity. This is a *1:N* relationship.

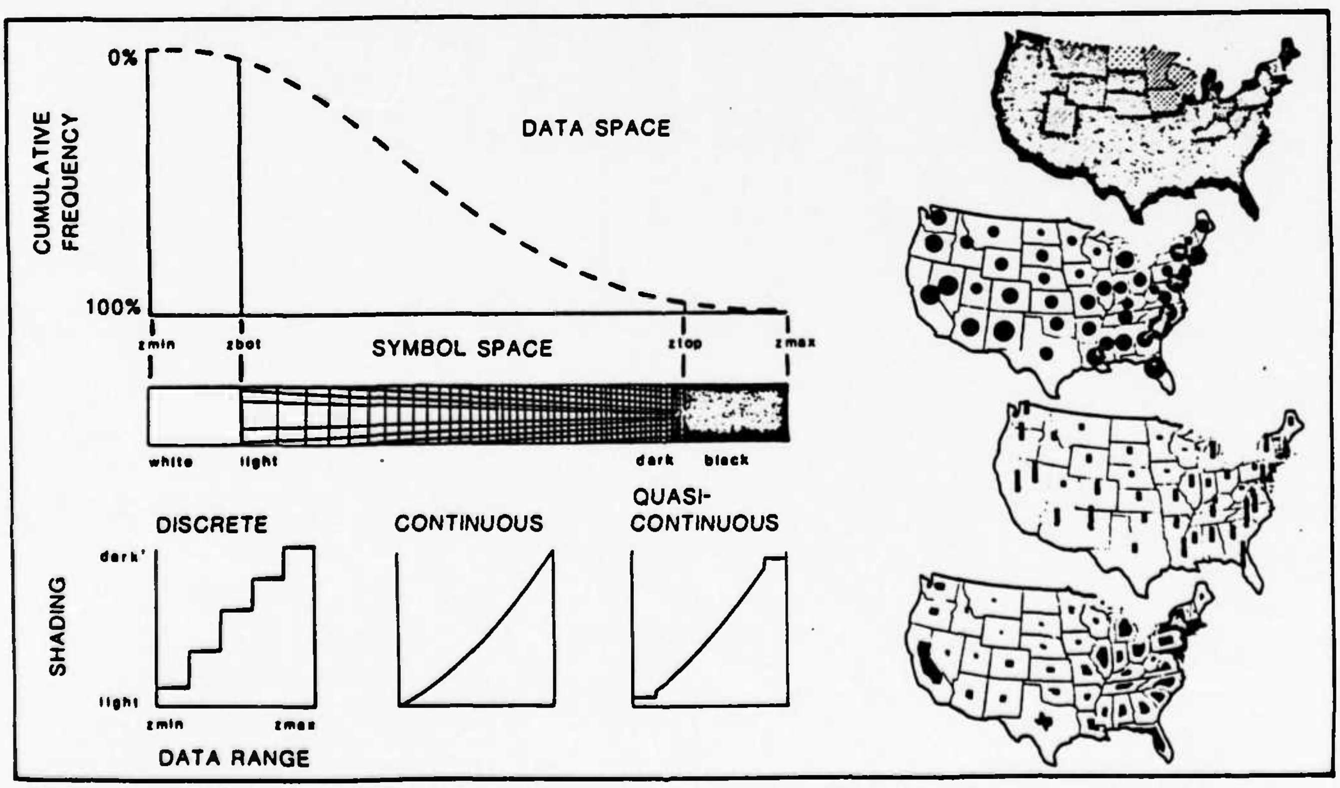
The primary shading algorithm used by MAPSOFT to generate choropleth maps uses a continuous rather than class interval classification method. The minimum and maximum data values in the distribution of the variable to be represented are first determined, and these extremes are assigned the display values of WHITE and BLACK. Intermediate values are then assigned a shading density proportionally between the black-white extremes according to a nonlinear psychophysical power function. A similar scheme is used to calculate the symbol sizes for circles, bars, and polygons on graduated-symbol maps.

When the frequency distribution is highly skewed, a quasi-continuous allocation variation is used instead. This involves setting data value cutoff points for WHITE and BLACK display, and shading only values in the intermediate range. Quasi-continuous shading represents a compromise between continuous-tone and

MAPSOFT: Interactive Graphic Interface Design



Screen Presentation Showing Elements of the Map



Display Options for Data Distribution Presentation

Source: Utano, "A Portfolio of Computer Mapping Software at Akron University"

the more traditional discrete classification interval approach. It is a useful interpretation tool, but can introduce perceptual bias if it is applied to inappropriate types of data.[43]

The processing sequence used by MAPSOFT consists of the following steps:

1. Initialize default parameters.
2. Read global parameters.
3. Read, echo, and check user-supplied local parameters.
4. Read the geographic and statistical data.
5. Determine minimum-maximum geographical/statistical data values.
6. Calculate appropriate symbolism for the areal units.
7. For each areal unit:
    1) Read in the x-y coordinate boundary.
    2) scale the coordinates to plotter space.
    3) draw the appropriate areal symbolism.
8. Draw the legend, cumulative frequency curve, main and subtitles, and the surrounding line.

7.8. NIMGRID and ODYSSEY

The GRID and IMGRID multivariate spatial analysis systems were developed at Harvard in the 1970's. They organize the database as a collection of multiple layers, with each layer representing a specific variable over a fixed area. Each of these variables is treated as a separate, registered array of some machine-dependent maximum size and all of the arrays are stored in a single random access file. A major drawback to this design is the existence of a fixed limit, however large it might be, on the size of the grid data arrays. Processing which involves a variable also requires its complete array to be resident in memory, even though only a fraction of it might be accessed concurrently.

The newer NIMGRID system resolved both these problems by modifying the data arrangement from a conventional grid to a raster-oriented grid arrangement.

Each variable is represented as an image, which allows the use of raster image-processing methods to process it on a line-by-line basis.[30]

More recent work at Harvard has focused on integrating, under the ODYSSEY system, the various systems developed there over the past fifteen years. ODYSSEY's basic structure is the *least common geographic unit*, a polygon formed by overlaying and cutting all boundary point chains bounding any region in the database. The resulting subregions contain constant data values. Collectively, these units constitute the basic location set managed by the system and determine the maximum resolution.

7.9. NURE (National Uranium Resource Evaluation)

A sediment and water analysis database system used by the Savannah River Laboratory collects and analyzes data on the basis of National Topographic Map Series (NTMS) units. These are 1 by 2 degree quadrangles, and the data for each is kept as a separate data set. The SAS software package allows these separate data sets to be concatenated and considered as a single unit for plotting areas extending over multiple quadrangles. [19]

The NURE database project demonstrates that it is possible to take advantage of existing commercial graphics and statistical display packages to display data previously selected on the basis of location. This proves the feasibility of developing a low cost yet powerful GIS by concentrating on operations involving only the spatial characteristics of data. The results of such selection and manipulation operations can then be submitted to these existing packages.

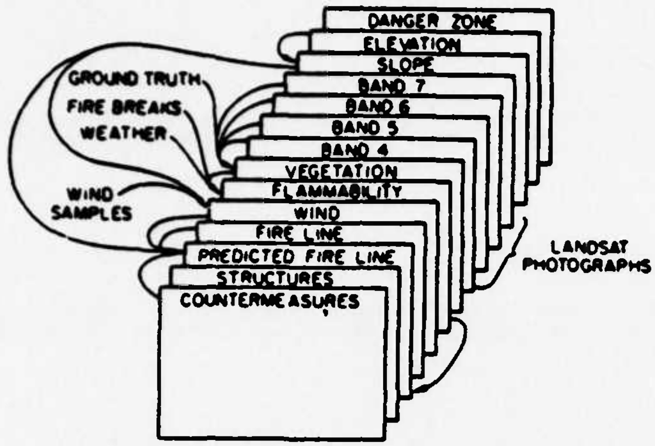
7.10. PICDMS

The logical structure of PICDMS is unique in that it does not conform to any of the standard approaches. Rather, it has adopted the dynamic stacked-image as its fundamental logical structure. The model is suitable for use in general pictorial database applications and provides flexibility in representing relationships and semantic information.

The dynamic stacked image structure is an extension into three dimensions of the grid data structure. Each stack structure consists of a set of two-dimensional frames. These may be viewed as registered grid cell arrays, all of which share common dimensions. Each stack (there will be one for each region managed by the system) functions as a collection of logical records whose format varies as image grid arrays are added to, or removed from, the database.[4] All values for a given region and grid cell are stored in corresponding positions of the appropriate frames.

The dynamic stacked-image model differs from more conventional database design methods in that the record structure representing a geographic region is itself a variable. Adding a new image is equivalent to adding a new field to this record. Conceptually, a stack may be thought of as a region or polygon. Adding a new image to an existing stack is then equivalent to simultaneously adding a new attribute to the region and storing the relationships between that attribute and any others defined over the same region. The LDMS concept of location data sets is somewhat similar in that each LDS is in effect registered to the entire global region.

Adding Derived Images In PICDMS



Arrows show derivation of images from other images, and from non-image data, e.g., the WIND image derived from the SLOPE image and non-image WIND SAMPLES.

A brushfire database.

Data Manipulation Commands to Add a Slope Image:

```
ADD (IMAGE SLOPE CHAR (5))
   IF (ELEV(I-1,J) < ELEV(I,J))
        & (ELEV(I,J-1) < ELEV(I,J))
        & (ELEV(I,J + 1) < ELEV(I,J))
        & (ELEV(I + 1,J) < ELEV(I,J))
     THEN SLOPE = 'PEAK',
   ELSE IF (ELEV(I,J-1) > ELEV(I,J))
        & (ELEV(I,J) > ELEV(I,J + 1))
        & (ELEV(I-1,J) = ELEV(I + 1,J))
     THEN SLOPE = 'SOUTH',
   ELSE . . .,
   FOR ZONE = 'DANGER';
```

PICDMS slope calculation.

Source: Chock et al, "Database Structure and Manipulation Capabilities of a Picture Database Management System"

PICDMS is specifically intended to integrate image and numeric data. Furthermore, the designers envision evolution of PICDMS into a unified database system in which the image or non-image nature of data is transparent.[4] Grid registration directly stores shared location relationships between image and variable data in the same stack. Specification of the coordinate system relating individual grids is performed as part of the logical database definition process before any data is put into the image stack. This process includes stipulation of scale information in terms of the logical length and width of the grid arrays; these values may be given in lat/long coordinates (or fractions thereof), meters, or feet.

The PICDMS data manipulation language provides the flexibility to specify at least twenty-six distinct classes of operations on images, points, lines, regions, and their attributes. The freedom to define grid variables over a wide variety of data types permits identification of coordinate positions for any values stored in any grid in the registered stack. Furthermore, new grid images may be computed, added to a stack, and themselves used as the basis for retrieval operations.

7.11. POLYGRID

The POLYGRID system represents an effort to capitalize on the advantages of both grid-based and polygon-based systems. The grid mode offers advantages in performing data manipulations, analysis, and composite mapping. The advantages of polygon mode lie in its superior resolution and flexible display capabilities, including manipulations of scale, projection, and orientation.

POLYGRID uses a software package called GRIDCHAIN to identify the grid cells corresponding to grid region boundaries and to produce a corresponding

image/center file. This derived file can then be displayed using the polygon mode. Conversion between the two formats involves the use of decision rules which have been statistically validated to produce acceptable correlation. [20]

The POLYGRID system does not appear to represent a true hybrid in that the source data remains stored in grid form. The image/center files represent redundant copies of this data which has been processed to facilitate vector-oriented output operations. While the database may include an assortment of files in both formats, the term hybrid would seem to imply that data is stored in a form suited to both raster and vector operations, which is not the case.

7.12. REAP (Regional Environmental Assessment Program)

An integrated data management system developed for North Dakota's Regional Environmental Assessment Program combines a conventional database management system with map and graphic display capabilities. The DBMS serves as the central unifying component and is used to manage polygon, grid, cellular, and alphanumeric data. The maps are stored in the database as polygon, line, and point plotting codes. Several standard commercial graphics and statistical packages, such as SPSS, DISSPLA, and TELL-A-GRAF are also integrated into the overall design. [14] This appears to be a very well designed system, developed to support eight general categories of user requests:

1. Conventional queries and report generation.
2. Maps of various scales and projections.
3. Statistics, especially trend analysis.
4. Charts and graphs.
5. Numeric calculations, for example of simple engineering problems.
6. Composite mapping.
7. Basic geographic capabilities, such as area computation.
8. Models, to answer "What if...?" questions. [14]

Collections of data are designated as *titles*, and the DBMS maintains a directory of these. The system recognizes two classes of titles, spatial and alphanumeric. Each spatial title may be associated with several alphanumeric titles, but alphanumeric data titles are linked to only one spatial title. Restated, this means that a given type of alphanumeric data is collected and stored at only one level of a region hierarchy.

The QUERY language of REAP includes constructs for expressing relationships between spatial data titles and alphanumeric data titles. A MAP subsystem draws maps on either graphics or plotting devices, and is fully integrated with QUERY. For example, a query which produces as its result a list of map regions or polygon names may serve as the input to MAP. [14] LDMS provides a similar capability, in that a list of geographic entity names produced by conventional database data manipulations can be submitted to LDMS for display.

7.13. SHADE (Simple Handling of Areal Data Expressions)

SHADE uses the bounding rectangle strategy to quickly screen polygons according to location. It also uses a tree-structured file system but, where LDMS separates region locations and attribute data, SHADE stores this information together.

SHADE is the only system identified which specifically provides for distribution of data values over the area of regions, a concept central to the location data set approach of LDMS. The method involves the uniform distribution of data values over a gridded equivalent to the polygon region involved. The data are then allocated to overlapping polygon regions according to the proportion of areal

overlap. [42] The system also provides an interactive facility through which the user may define the location of polygons and points.

7.14. STORET

STORET, a water quality database used by the U.S. Environmental Protection Agency, provides facilities for interactive definition of regions known as *standard zones*. Users may specify values for attributes to be stored with these definitions. Zones are displayed over base maps retrieved and composed from three separate map databases:

1) states and counties,
2) hydrological features
3) municipalities.

Retrieval from the databases is keyed on latitude/longitude; coordinate extremes of every feature are stored in a separate directory to support fast identification of features falling within a search window. [34]

Several features of STORET are applicable to the design of LDMS. These include the bounding rectangle strategy, an interactive region definition capability, the base map overlay concept, and separate organization of region and map definitions.

7.15. USGS (U.S. Geological Survey)

As might be expected, storage and retrieval of cartographic information is a central concern of the USGS. It has therefore developed a number of different systems and file structures to support its requirements.

The digital cartographic files produced as part of the National Mapping Program (NMP) are of two basic types. Digital elevation model (DEM) files consist

of elevation samples; digital line graph (DLG) files contain planimetric map information on basic data categories such as transportation, hydrography, and boundaries. A separate but related Geographic Names Information System contains information on all names of geographic and other features that appear on USGS topographic maps. The entries include the type of feature, location by latitude and longitude, and the name of the quadrangle map on which it appears. [27] Both quadrangle maps and name files could be directly utilized by LDMS if provisions were made for interpretation and display of the graphics primitives which they contain.

The U.S. Geological Survey makes digital cartographic map files available through the National Cartographic Information Center (NCIC). A program is currently underway to provide a uniform, consistent digital cartographic data base for the coterminous U.S. by the early 1990's. When completed, it will consist of 54,000 7.5-minute quadrangles at an accuracy equivalent to that of 1:24,000 topographic maps. [27] At the present time, three types of topographic data are available from the NCIC:

1. Digital Terrain Tapes (DTT's) digitized at the resolution of 63.5 meter grid cells.
2. Digital Elevation Model (DEM) format tapes in which the data are organized in cells of 3-arc seconds (about 100 meters) rather than a grid based on the Universal Mercator projection.
3. Digital topographic data available as a by-product of photo generation. These are of relatively higher resolution and accuracy than the first two types, and are available for many 7.5-minute quadrangles in DEM format. [15]

The U.S. Geological Survey is also developing a nationwide, multi-purpose digital cartographic database of land use and land cover data. The classification system involves nine general categories and these are further subdivided into a

United States Geological Survey
Land Use and Land Cover Categories

LEVEL I	Level II
1 Urban or Built-up Land	11 Residential 12 Commercial and Services 13 Industrial 14 Transportation, Communication, Utilities 15 Industrial and Commercial Complexes 16 Mixed Urban or Built-up Land 17 Other Urban or Built-up Land
2 Agricultural Land	21 Cropland and Pasture 22 Orchards, Groves, Vineyards, Nurseries 23 Confined Feeding Operations 24 Other Agricultural Land
3 Rangeland	31 Herbaceous Rangeland 32 Shrub and Brush Rangeland 33 Mixed Rangeland
4 Forest Land	41 Deciduous Forest Land 42 Evergreen Forest Land 43 Mixed Forest Land
5 Water	51 Streams and Canals 52 Lakes 53 Reservoirs 54 Bays and Estuaries
6 Wetland	61 Forest Wetland 62 Nonforested Wetland
7 Barren Land	71 Dry Salt Flats 72 Beaches 73 Sandy Areas other than Beaches 74 Bare Exposed Rock 75 Strip Mines, Quarries and Gravel Pits 76 Transitional Areas 77 Mixed Barren Land
8 Tundra	81 Shrub and Brush Tundra 82 Herbaceous Tundra 83 Bare Ground Tundra 84 Wet Tundra 85 Mixed Tundra
9 Perennial Snow or Ice	91 Perennial Snowfields 92 Glaciers

Source: Guptill, "Thematic Map Production from Digital Spatial Data"

total of thirty-seven subcategories. The data is encoded in vector format and would need to be converted to raster grid format for direct use by LDMS. The USGS has already developed a polygon-to-grid (PTG) program to perform the conversion to a run-encoded raster format. The conversion is claimed to be very efficient, requiring less than 3 minutes of computer time to convert the data to be plotted from its polygon/arc segment format into raster format. [16]

8. Implementation Strategies

8.1. System Overview

LDMS includes built-in checkpointing mechanisms that save the results of major processing phases in intermediate files. The content of these files is expressed in high-level, logical terms to enhance modularity. Because of this convention, LDMS is suitable for use as a semi-autonomous module operating in cooperation with a conventional database management system. Both selection criteria and transaction results could be passed between the two in the form of entity names or attribute values.

Many of the operations most frequently performed by LDMS involve sets. This includes especially the selection of distinguished locations on the basis of their region membership or the data values associated with them. Because the grid model is well-suited to such operations, it was selected as the underlying logical framework. Storage requirements, always a potential drawback of the grid model, are minimized through careful design of the major structural elements. At the same time, the software components of LDMS exploit the combination of program segmentation, interactive analysis, and raster data organization which

make the gridded database structure a most powerful planning tool.[30]

Structural Elements

The number of different structural elements in LDMS is purposely limited in an effort to simplify the design and promote modularity, flexibility, and generality. Spatial entities, location data sets, and two distinct types of pseudofiles were judged to be the minimum essential elements.

All geographic entities are managed as regions. A Region Directory file contains a name and location predicate pair for each region which has been explicitly declared to the system. LDMS treats each name and its location predicate as being totally equivalent. Location predicates are encoded using a combination of the bounding rectangle strategy and autoadaptive block coding. The bounding rectangle is specified in whole degrees of latitude and longitude. The autoadaptive coding string then describes more precisely the location of the entity within that rectangle.

Location data sets are organized as a quadtree variant that uses a sixteen-way fixed regular decomposition. Only those subtrees for which actual data exists are developed so as to minimize storage requirements. For some types of data these trees may nevertheless assume the form of fully developed but sharply tapering pyramids.

Input and output operations adopt a pseudofile approach. Two types of standardized files are involved: *submit files* and *display files*. The first of these corresponds to a formatted version of the user's request and the second represents a high-level expression of the final result. This method is compatible

with the development of an interactive graphics user module as the system's front-end. At the same time it provides the flexibility and modularity which would allow other types of interfaces to be fitted easily to LDMS.

Software Components

With the exception of a few utilities, all LDMS software components may be conveniently partitioned into four groups according to the types of operations they perform on pseudofiles. These four groups are the Interface System, the Request Processing System, the Data System, and the Display Processing System.

The *Interface System* (IS) is concerned with interpreting the intentions of the user and creating a *submit* file to express them. The primary interface uses a commercial device-independent graphics package.

The *Request Processing System* (RPS) is responsible for interpreting a previously-created *submit* file and producing a *display* file corresponding to the result. Although it may request data values from (or supply them to) the Data System in the course of processing, the responsibility for performing logical operations on that data remains solely with the RPS. The RPS is also tasked with updating the system's Region Dictionary. Note that data associated with an entry which is deleted from the Region Dictionary may be retained in the system and still be available for reference and manipulation. This is made possible by the total separation of data and entities which is central to the LDS approach.

The *Data System* (DS) is solely responsible for performing operations on the Location Data Sets. This includes data retrieval, update, insertion, and deletion. All of these are performed so as to maintain the consistency of each LDS in

accordance with its declared semantic data class.

The *Display Processing System* (DPS) interprets a previously-created *display* file and presents it to the user. The actual output format will be determined by the context options currently selected by the user, the output device type, and the content of the *display* file itself.

LDMS requires very little in the way of original algorithm development. Most of the basic functions and algorithms may be adapted from those used by other grid-based systems. Dangermond provides a good description of the data manipulations performed by GIS systems and how they are implemented on both gridded and topological structures.[8] Variations of most of these which are suitable for use with quadtrees have also been previously published.[41,39] Some algorithm extensions are necessary in the area of integrity constraint, however. Most of these involve operations on region centroids, initial distribution of values over a region grid, and the redistribution of these values during subsequent updates. The basic strategy is to distribute data values over the area occupied by the update entity according to a proportional fit criterion. The method provides for differentiation between subregions which are available for data redistribution and those which cannot be modified without introducing potential data consistency errors.

8.2. Flow of Control

The top-level processing algorithm used by LDMS is similar to that employed by MAPSOFT. It includes the following steps:

Flow of Control in a Location Data
Management System (LDMS)

```
  +--------+  +-+      +--------+      +-+   +---------+
  |Inter-  |  | |      |        |      | |   |Display  |  +---------+
->|  face  |->| |      |Region  |      | |-> |Proc-    |->|Device A |
  |   1    |  | |      |Diction-|      | |   |  essor  |  +---------+
  +--------+  | |      |  ary   |      | |   +---------+
              |S|      |        |      |D|
  +--------+  |U|      | [file] |      |I|   +---------+
  |Inter-  |  |B|      +--------+      |S|   |Display  |  +---------+
->|  face  |->|M|          ^           |P|-> |Proc-    |->|Device B |
  |   2    |  |I|          |           |L|   |  essor  |  +---------+
  +--------+  |T|          |           |A|   +---------+
              | |          |           |Y|
  +--------+  | |          |           | |   +---------+
  |Inter-  |  |F|     +----|----+     |F|   |Display  |  +---------+
->|  face  |->|I|     |         |     |I|-> |Proc-    |->|Device C |
  |   3    |  |L|---->| Request |----->|L|   |  essor  |  +---------+
  +--------+  |E|     |Processor|     |E|   +---------+
              |S|     |         |     |S|
  +--------+  | |     +---------+     | |   +---------+
  |Inter-  |  | |          ^           | |   |Display  |  +---------+
->|  face  |->| |          |           | |-> |Proc-    |->|Device D |
  |   4    |  | |          |           | |   |  essor  |  +---:-----+
  +---:----+  | |          |           | |   +---:-----+      :
      :       | |          v           | |       :            :
  +---:----+  | |     +--- ------+     | |   +---:-----+      :
  |Inter-  |  | |     |   Data   |     | |   |Display  |  +---:-----+
->|  face  |->| |     |  System  |     | |-> |Proc-    |->|Device Z |
  |   N    |  | |     |          |     | |   |  essor  |  +---------+
  +--------+  +-+     +----+-----+     +-+   +---------+
   +---------------+-------|-------+---------------+---------------+
   |               |               |               |               |
   v               v               v               v               v
  +-+             +-+             +-+             +-+             +-+
  |L|             |L|             |L|             |L| . . . . . . |L|
  |O|             |O|             |O|             |O|             |O|
  |C|             |C|             |C|             |C|             |C|
  |A|             |A|             |A|             |A|             |A|
  |T|             |T|             |T|             |T|             |T|
  |I|             |I|             |I|             |I|             |I|
  |O|             |O|             |O|             |O|             |O|
  |N|             |N|             |N|             |N|             |N|
  | |             | |             | |             | |             | |
  |D|             |D|             |D|             |D|             |D|
  |A|             |A|             |A|             |A|             |A|
  |T|             |T|             |T|             |T|             |T|
  |A|             |A|             |A|             |A|             |A|
  | |             | |             | |             | |             | |
  |S|             |S|             |S|             |S|             |S|
  |E|             |E|             |E|             |E|             |E|
  |T|             |T|             |T|             |T|             |T|
  | |             | |             | |             | |             | |
  |a|             |b|             |c|             |d| . . . . . . |z|
  +-+             +-+             +-+             +-+             +-+
```

1. *Read global parameters*. This involves referencing a set of program context variables which specify the defaults and assumptions to be used by the system when interpreting user requests. Most of these variables default to system-supplied values but may be reset by the user at any time. They include the scale, current definition of neighborhood, the implied scope of commands, and display options such as format and color.
2. *Read, echo, and check user-supplied local parameters*. This would be performed by the Interface System during creation of the submit file.
3. *Read the geographic data*. In LDMS, these functions would be initiated by the RPS or performed by it on behalf of the Interface System. For example, the IS might reference the Region Directory file to determine the location predicate of a named regions or validate its existence. The Request Processor would reference the location data sets (through the Data System) for two reasons. First, to determine the data values corresponding to specified location predicates and second, to determine the location predicates (distributions) of specified data values.
4. *Determine minimum-maximum geographical/statistical data values*. Both the Request Processing System and the Data System would share responsibility in this area. The RPS performs all logical operations involving selection by location and the DS processes selections against data stored in the Location Data Sets. The bounding rectangle portion of the location predicate makes determination of geographical extent a trivial operation. Determination of attribute extremes may be more complex. For some data types, exploitation of inheritance or propagation properties of the semantic data class and judicious pruning of the LDS subtrees will help to reduce processing costs. In other cases, there may be no alternative to an exhaustive search. In general, however, most searches will likely be restricted by limits on the data values, world area, or both.
5. *Calculate appropriate symbolism for the areal units*. In LDMS, the designated primary base map set will serve as the output default. The appropriate quadrangles are selected as a function of the location predicates specified either directly (named regions) or indirectly (extent of attribute value distribution) in the query. This will involve little or no calculation. The user has the option of selecting from among several base map sets or graphic display options using the context menu. Graphic display options might include minimum polygon coloring, choropleth maps, etc.
6. *Display the area or data corresponding to the query result* For interactive map display, scaling is performed automatically by adjusting the limits of the world coordinate window on the output device.[11] This window is by default set to the smallest bounding rectangle that includes the location predicate of the query result. Alternatively, the user may force a series of displays to the same scale by declaring a display scale or region as part of the current context. In either case, it will usually be necessary to extend the bounding rectangle slightly in either the X or Y direction so as to match its aspect ratio to that of the screen's viewing window. This will prevent unpredictable distortion of the image and produce a standard Mercator projection of the region

of interest. An alternative would be to adjust the aspect ratio of the bounding rectangle as a function of its mid-latitude. This would reduce distortions of area and scale in the region of primary interest, albeit at the cost of introducing errors in angular relationships; in cartographic projection, nothing comes free.

LDMS processing modules create a *display file* which may be interpreted by various display modules for output to the CRT, line printer, plotter, or other output devices. This file defines the locations of interest and the world coordinate window (superregion) within which they lie. The initial implementation includes just the CRT output module. It references and displays, in mosaic form, the quadrangle base maps required to produce a composite map of the complete world coordinate window. The direct plotting of location predicates as colored approximations to polygon regions and the tabular or symbolic display of numeric data are other output options.

7. *Draw the legend, main and subtitles, and any required labels.* In LDMS, much of this information will be directly displayed on the output map itself. Labels of derived regions would be applied at the centroid of their individual bounding rectangles. Due to this bounding rectangle display strategy, the scale of composite maps may vary greatly from one query to the next. It will therefore be important to provide a reference framework with each final map, in the form of either a distance scale or latitude-longitude graticule.

8.3. Geographic Entities

A difficult design problem concerns the desirability of separating entities into classes based on their form. Geographic entities are conventionally divided into *point*, *line*, and *region* types according to their geometry. To some extent, such distinctions are arbitrary in that it is possible to define or encode each of these in terms of the others. [29] Nevertheless, most geographic information systems do distinguish between them in the interest of processing and storage efficiencies and better location precision. Unfortunately, categorization of entities in this manner carries with it certain restrictions and implicit assumptions. These in turn may limit not only the representation of geographic entities, but the types of operations which may be applied to them as well. One of the most serious consequences is that similar entities may be placed into different categories solely on

the basis of scale considerations. If Bear Lake is classified as a point, what about Lake Michigan? Similarly, how would one calculate the total incorporated urban area of a state if its larger cities are represented as regions and its smaller towns as points? Is the Mississippi River, which extends over an area larger than many counties, a line or a region?

The user may, of course, establish guidelines to reduce the number of such inconsistencies. Unfortunately, doing so has the effect of setting arbitrary resolution limits on the location specifications of geographic entities. This limitation would be especially serious under the LDS approach, where attribute values are derived through the entity's location rather than being explicitly stored with the entity.

In light of the foregoing considerations, all geographic entities managed by LDMS are treated as regions. This approach recognizes that points and lines are merely geometric abstractions; all locations in the real world occupy space. Thus, in lieu of points, LDMS deals with "small regions"; in place of lines, "linear regions". All regions are represented by organizing their interiors rather than specifying their boundaries. This avoids any need to treat disconnected or multiply-connected regions as special cases.

8.4. Location Predicates

Location predicates serve two purposes in LDMS. At one level, they represent an encoded form of the region's location and depiction. However, they also function as logical links through which multiple entities may be directly associated with the same entry in a given Location Data Set. Location predicates use

a variation of autoadaptive block coding modified for compatibility with the bounding rectangle technique. We will refer to it as *adaptive resolution*. The effect of adaptive resolution is to force all region definitions to be of fixed length. This simplifies design of the data structures and limits the storage required to represent large or complex regions. As Chock has pointed out, the alternative method of selecting a single grid resolution for all cases is very wasteful of storage. Use of a small cell size to obtain satisfactory resolution for detailed or small features produces a high level of redundancy when it is applied to larger or less detailed objects.[5]

Another factor which weighed heavily in the design of a location predicate representation was the nature of a CRT graphic output display. In the case of cartographic systems, all data is collected and stored at the lowest resolution possible. This is especially true of location and shape information because there are no inherent limits on output map size and scale combinations. Clearly, this is not the case with an interactive system which uses a fixed-size display. This arrangement simply does not permit geographic entities of large extent to be viewed at the same resolution as smaller ones. To put it another way, the maximum possible display resolution is inversely proportional to the real world size of the entity. This does not preclude subdivision of large regions into smaller sections, but these then become regions in their own right and the scale-size relationship still holds.

As a consequence, there is no need to encode the location of large regions at the same resolution as smaller regions or point entities. This is significant because both grid and topological encoding schemes generally require more storage to represent large irregular regions than smaller ones defined at the same resolution.

Adaptive resolution provides a means by which to offset these larger storage requirements by reducing the resolution of larger regions. This means that a fixed amount of storage is sufficient to encode the location of any size entity. This amount may be set by the system administrator based on display device characteristics and available storage.

The notion of a *bounding rectangle* is encountered frequently in geographic information systems. This involves encoding the maximum and minimum coordinate values associated with an entity or feature. Its popularity is due to the efficient manner in which it supports determination of inclusion relationships. In the general case, this may involve checking several regions and testing the extent of each. Because operations involving inclusion are among the most important and most frequently encountered, the bounding rectangle strategy has been adopted in the definition of location predicates.

The approach of LDMS is to subdivide the area of the bounding rectangle into as many fixed-size blocks as the number of binary digits allocated for first-level region resolution will permit. The digits allocated for secondary resolution are then used to expand the GRAY blocks among these. As in the basic version of the autoadaptive coding technique, WHITE blocks within the bounding rectangle are represented as *0's* and BLACK ones as a *1*. In the LDMS design, however, GRAY blocks are not necessarily always represented by the digit *1*. Instead, GRAY blocks are represented by the digit which is *least* representative of the rectangularly bounded region. This convention increases the information content of the binary string, making it possible to quickly determine the absolute presence or absence of the region from most of the rectangular area. Thus, a

rectangle enclosing a predominantly diagonal linear region would be designated a *sparse* region and *0's* would correspond to WHITE blocks; both BLACK and GRAY blocks would be encoded as *1's*. In contrast, a rectangle enclosing a predominantly horizontal or vertical region would likely be encoded as a *dense* region, with *1's* indicating only BLACK subblocks.

Many data manipulation and query processing operations can be completed using only the bounding rectangle and primary resolution bits of the location predicate. Regions, for example, are usually displayed as sections of base maps rather than direct plots of the location predicate. The secondary resolution bits are therefore used primarily to ensure an acceptable fidelity when defining derived regions in terms of other regions.

8.5. Location Data Sets

Location data sets are implemented as data structures with characteristics of both multidimensional trees and pyramids. Like quadtrees, they involve a regular decomposition of the global region represented by a root node. Also like quadtrees, only quadrants for which data is available at the next lower level are further developed. However, certain modifications have been implemented in an effort to maximize the fanout factor at each level of the tree. There are also provisions for multiple-page nodes where required.

Like pyramids, data applicable to all valid scales is physically stored in the data structures rather than being derived from base-level data on an as-required basis. While that other alternative would have lower storage requirements, it would also increase both processing and input/output costs by forcing the system

to access leaf data nodes for virtually every operation. Therefore, it was deemed advantageous to store data in both internal and leaf nodes. Each such node represents a distinguished location as determined by data insertions. The basic strategy is to associate data values with the locations of the geographic entity for which it is supplied. This data is pushed down into the quadtree structure to a level at which the entire data quadrant is contained within the extent of the update entity. To avoid ambiguity, LDMS only updates location data sets on the basis of the BLACK distinguished locations of the entity.

New location data sets may be defined to the system at any time, similar to the way in which PICDMS allows frames to be dynamically added to a region stack. In LDMS, a system-maintained dictionary of currently-defined Location Data Sets defines the content of the current global stack. In contrast to the frames of PICDMS, however, Location Data Sets are dynamic; storage is allocated only to those subtrees and at those times required by data insertion.

Branching Factor Considerations

The worst-case fanout factor for each node is set at sixteen. The effect is as if each level of the structure possesses the resolution capability of two levels of a basic quadtree. To provide regularity in decomposition, the area represented by the root of each location data set extends beyond the limits of the earth's coordinate grid. Rather than make the fanout a program-defined constant, the number sixteen was selected for several important reasons:

1. *Representation Efficiency*. Virtually all computer systems are able to reference memory in terms of 8-bit bytes and many languages provide facilities for manipulating the individual bits. Several efficiencies are possible by encoding data on the basis of one bit per decomposition unit. Division of regions into sixteenths provides for a more regular decomposition than does the basic

byte. Furthermore, a single byte is adequate to specify any one of the 256 squares produced by a two-level region decomposition. This opens up possibilities for compressed encoding of "small" regions, the LDMS counterpart of point entities.

2. *Data Acquisition Compatibility*. Both one-degree latitude-longitude squares and the fifteen-minute squares which result from their decomposition into sixteenths represent something of a standard in geographic data collection and representation. U.S. Geological Survey Charts, for example, are drawn as fifteen-minute quadrant squares. Therefore, it is important that LDMS be able to reference units of those sizes exactly, without round-off error.
3. *Conversion Efficiency* A common base factor simplifies the mapping between location predicates and location data sets. Because location data sets are of fixed length, it is desirable to use a small value for the base. This reduces the minimum storage requirement of the Region Directory file.
4. *Minimize the Impact of Page Chaining* It is desirable to have as great a fanout factor as possible so as to increase the selectivity at each level during tree traversals. However, if pages must be chained into multiple page nodes to hold the data entries, then many of the advantages inherent in a tree structure are foregone. Given the desirability of a regular decomposition of regions, the possible fanout factors are 4, 16, 64, ...(N**2)**2. Sixteen was selected as the best choice, because many commonly-occurring page sizes cannot hold as many as 64 data value entries in addition to the fixed storage overhead of each node.

Semantic Data Class

Nagy and Wagle note that the operations which transform raw geographical data into meaningful information may be classified according to the characteristics of that data.[29] Their analysis assumes that all information will be attached directly to geographic entities and that it may be divided into *geometric* and *nongeometric* types of attributes. According to that typology, geometric attributes are those which specify the location and shape of the entity. The LDS approach also treats location, or more precisely location *sets* (because an entity may conceivably consist of several non-contiguous points or regions), as an attribute of geographic entities. It differs in that nongeometric attributes which fall into certain pre-defined semantic data classes need not *necessarily* be assigned to those

same entities. Instead, the user may designate them as *location data sets* and the system will automatically enforce those consistency constraints which follow naturally from the spatial nature of the data and its semantic data class. Major semantic data classes would include those displaying *inheritance, generalization, aggregation, stratification,* and *collection* properties.

Inheritance means that data values representing characteristics at one resolution level apply without variation to all lower resolution levels. Designation of the oceans as marine areas provides an example of this semantic data class.

Generalization implies that data values applicable to lower resolution levels (larger regions) represent a less-precise version of the values associated with the next higher resolution. Within the broad category of generalization, there are several important subclasses. These would include propagation of *average, maximum,* or *representative* values. Terrain elevation would be a candidate for treatment here, with propagation of either average or maximum values depending on the specific application. Land use, and other types of data in which the valid domain consists of a limited number of discrete values, provide examples of representative generalization. In such cases, it would be meaningless to average several numbers representing mutually exclusive and distinct categories. Rather, the predominant value might be selected for propagation to the next level. Generalization has many difficult practical problems associated with it; Nagy and Wagle, for example, pose the question of "How many trees make a forest?" [29]

Aggregation means that data values at any given resolution level represent the sum of the values which apply to subordinate regions. Population counts, mineral and ground water reserves, oil refinery capacity, and standing acre-feet

of timber resources are common examples.

Stratification is related to the concept of multi-scale entities. It is similar in some respects to generalization except that no loss of precision or distortion is implied. All resolution levels are equally valid, but it is not appropriate to deal with all of those levels at one time. Graphic depictions of a region are examples of this data type. Thus cultural or terrain features selected for display at lower resolutions represent a subsetting of those displayed at higher resolutions rather than a generalization of their image pixel intensities. Base map quadrangles, for example, may be specified for any level in the global decomposition, but would not be automatically maintained for all intermediate levels due to their storage intensive nature. Character string labels and symbols representing features of interest could be similarly treated.

Collection refers to data values which remain individually distinct but which are collectively propagated from one level to the next. In some respects, it is a form of reverse inheritance in that regions acquire the properties of all of their subordinate entities. It differs primarily in that multiple data values may be involved. Examples of the collection semantic data class typically involve lists of characteristics or features, such as languages spoken or the names of native fauna and flora. Data values frequently take the form of item or object identifiers: report numbers, names of businesses, legal subunits, etc.

8.6. Input/Output Format

The data formats required by input and output display devices may differ from internal data formats selected for processing and storage efficiency. There-

fore, modularity is enhanced by localizing the necessary conversion routines in the input-output processing modules. LDMS has adopted an intermediate pseudofile approach for this reason. The results of major processing phases are stored in logically formatted files for subsequent interpretation by other modules. The information contained in these files may thus be converted during processing into whatever format efficiency, user preference, and hardware considerations dictate. The same output display file, for example, could be processed by vector, raster, or printer device display modules. This design also allows pseudofiles representing frequent queries to be stored for repeated execution, thereby bypassing a considerable amount of repetitive intermediate processing. The pseudofiles used by LDMS are of two types:

Submit Files. These correspond to either update or retrieval requests. Each record entry includes a *control* field, reserved for future use; a *label* field which specifies the name (if any) of the entity; a *location predicate*, in standard format; a *data item* field which specifies the name (if any) of the data item represented, and a *value* field, which specifies the data value associated with the location predicate. Additional information, such as selection criteria and default display options, may appear in the file header.

Display Files. These correspond to the final result of query processing and are logical listings of entities to be displayed. Each entity record includes five fields: *control*, *label*, *location predicate*, and *value*. When LDMS is integrated with a conventional database management system, entity names and data values may be extracted directly from display files and passed to the DBMS for further processing.

Pseudofiles provide many of the benefits of processing checkpoints. Also, they promote independence between modules and avoid undue reliance on the features of any specific set of devices or graphics routines. Furthermore, because most forms of output possess a spatial component, they may be treated as entities in their own right and serve as the inputs for further processing.

8.7. Functions and Algorithms

The functions and algorithms of LDMS are those routinely provided by many existing GIS and DBMS implementations. Where they differ, it is primarily due to the need to adapt them for use with quadtrees and bounding rectangles.

Many of the basic algorithms for calculating geometric properties of images represented as quadtrees have been previously published. [39,40] Most of these are simple adaptations of basic tree traversals, differing primarily in the types of operations performed at the nodes. These may be easily extended to other than binary image data by modifying these operations. Indeed, the conceptual simplicity of the approach is one of the main attractions of quadtrees and related data structures. Because quadtrees represent a successive subdivision of region data, they are well-suited to identifying regions or points within specified distances of each other. Distance search is an important function in a GIS because many queries can be expected to involve selections based on proximity relationships. Its calculation requires a determination of the distance from each cell to the nearest occurrence of a specified value or class of values. Using this method, the proximity of a cell to a given area, line, or point may be calculated. [15] The quadtree form of location data sets speeds the calculation by permitting cells to be tested in blocks rather than individually.

To transform image data into quadtrees, a criterion must be chosen for deciding that an image is homogeneous (i.e., uniform). One such criterion is that the standard deviation of its gray levels is below a given threshold t. The case where $t = 0$ is a special case which corresponds to the exact representation of an original image. [39]

The bounding rectangle strategy not only facilitates determination of inclusion properties, it also simplifies determination of the approximate centroid of a region. This is useful for operations such as label placement. [29]

The conversion of user-defined regions into location predicates, either through a graphic interface or definition in terms of previously-defined regions, is a potentially expensive operation. Fortunately, it need only be performed once per region, and then only for regions intended to be referenced again in the future. The method chosen is an adaptation of one described by Samet for converting a binary array into a quadtree representation. [39] It involves loading a scaled representation of the region into a boolean array of higher resolution than any conceivable final encoding. The bounding rectangle of the region is then subdivided and converted to a fixed-length binary string representation through a process of subdivision and merger of homogeneous blocks.

A consequence of the different base used to encode location predicates and location data sets is that some conversion mechanism must be provided. Because the one-degree grid square is common to both the bounding rectangle (LP) and global (LDS) based tree structures, the conversion is relatively straight forward. Furthermore, mapping between the two can be deferred until a final LP is required.

Logical Operations

Any geographic information system must provide a certain minimum set of application specific functions. Frequently tested relationships are usually classified according to the geometry of the entities involved. [29] While LDMS deals

exclusively with entities of type *region*, the need for equivalent function remains. We will therefore speak in terms of relations and logical operations involving *points, lines,* and *regions*. There are six possible combinations of these:

1. **Point-Point Relations.** These include *coordinate conversion,* which involves calculation of alternative cartographic projections as well as determination of equivalent positional notations in different reference systems. Another important relation concerns the *identity* of a point. When a previously-defined entity is entered again in a slightly variant form, the system must be able to recognize their equivalence.

2. **Point-Line Relations.** The primary operations in this category involve identification of intersection points. Where networks of line segments are involved, shared endpoints and intersections may represent *nodes* of particular interest.

3. **Point-Region Relations.** The most important operation here is that of *inclusion:* Does the given point fall within the specified region? It also includes the capability of identifying all (overlapping or nested) regions which contain the point. Determination of the *centroid* of a region also falls within the point-region category. Another variation involves the determination of the *nearest neighbor* of a specified point, where that neighbor may be either one of a set of simple points, or part of a line or region entity.

3. **Line-Line Relations.** The *identity* relation applies here. Calculation of line *length* may be very important as well as very costly to compute. This is especially true in systems like LDMS which are ultimately based on the grid data model. Determining the length of highly convoluted curves, such as those

representing mountain streams, is complicated by apparent length reductions produced by data sampling and resolution constraints. [29]

4. **Line-Region Relations.** Determination of *inclusion* relationships is a common operation in this category. For example, does a certain highway cross any desert regions? Construction of regions defined in terms of linear bounds may also be important. In LDMS, this last operation is facilitated by the bounding rectangle positional representation.

5. **Region-Region Relations.** The *area* of a topologically defined region is usually computed by integrating along region boundaries with respect to one of the coordinate axis. While that is an efficient method when regions are defined by their outlines, regular decomposition methods which organize the interior of a region are amenable to a summation approach. This is the method used by LDMS. Organization of interiors also eliminates the need to treat *islands* or multiply-connected regions as special cases. Logical operations, too, are facilitated by this hierarchical organization. These include calculation of the *intersection, union,* and *difference* of regions.

Location Data Sets

A natural consequence of the treelike structure of quadtrees is that many basic operations may be implemented as tree traversals. [39] Locating adjacent or closest neighbors is an important operation, and one supported well by quadtrees. The basic method is to ascend the tree until a common ancestor with the neighbor is located, and then descend back down the tree. Locating adjacent horizontal or vertical neighbors is straightforward; locating a neighbor in a corner (diagonal)

direction is a more complex but previously-solved problem. Adding pointers to each node to form *ropes* or *nets* simplifies the search, but at the cost of additional storage for the extra pointers. In practice, such measures are not necessary. [39]

Consistency Constraints

The automatic enforcement of data consistency across multi-scale entities entails an unavoidable increase in processing and storage overhead over systems lacking this capability. A portion of the storage overhead will be offset by accompanying reductions in storage redundancy for some data type, value, and spatial distribution combinations.

The use of a FIT parameter, set in the range of zero to one, governs setting and interpretation of fix bits which control redistribution of data values associated with a region. This is an essential function in order to ensure that some types of data values input on the basis of a large region are not mistakenly interpreted as belonging to much smaller regions without proper adjustment being made. One bit is required for each subtree, so that these 16 bits form part of the fixed storage overhead required by each node.

This FIT parameter also allows the user to set the degree of interpolation performed by the system. Thus, if the regions involved are strictly hierarchical, data values may be safely bound to the centroid of each region rather than being widely distributed. If, on the other hand, the system manages many different types of regions and these often overlap, then setting FIT to a high value assures that a small region which includes the centroid of a relatively much larger region will not mistakenly "inherit" the entire volume of data associated with the larger

region. This is accomplished by distributing the data of the larger region more widely.

9. Future Development and Extensions

Possible future enhancements which LDMS is designed to accomodate include the encapsulation of frequently-used data manipulation functions, the addition of indices, and development of improved efficiency measures.

An improved version of LDMS would encapsulate common data manipulation functions in a separate module and extend the data manipulation language accordingly. The modules would then assume responsibility for creation and execution of the *submit* files. Because the same grid structural model underlies both LDMS and PICDMS, it would seem that many of the PICDMS data manipulation algorithms could be adapted to this purpose.

The aggregate semantic data class would be suitable for maintaining indices of region names and their location predicates. Thus, one could maintain separate LDSs for counties, states, or sales regions to facilitate queries of the type "Show all counties with population > 20000."

Storage efficiency would be improved if compression and format conversion techniques were to be invoked automatically for certain specialized subcategories of regions. In addition, efficiency in both processing and storage would be improved by modifying Location Data Sets to support multiple-node pages. This could be accomplished by allowing pointers to descendent nodes to reference either a new page or an otherwise unused portion of the same page. Through such means, major subtrees (or even entire LDSs) for regions of sparse data

could be compressed into a single physical page.

10. Conclusions

This paper has outlined the design of a Location Data Management System which incorporates many features of existing geographic information systems. However, it differs markedly from them in that it adopts a Location Data Set approach as its underlying conceptual basis. The central idea of the Location Data Set approach is that spatial data should be directly associated with locations rather than named regions or points. The relationships between geographic entities and attribute values are in effect *derived* through the intermediate relationship of shared location rather than being explicitly associated with the entity. This approach represents a more accurate model of the real world than that used by most systems today.

While fundamental obstacles exist which must be overcome before LDMS could be fully implemented, a survey of related work has shown that none of these are insurmountable. In the final analysis, there are few relevant implementation issues which have not been satisfactorily dealt with in previous systems and few required functions for which the necessary algorithms have not been previously published. Data consistency enforcement algorithms are a notable exception, but the quadtree structure of Location Data Sets permits the extension of tree traversal methods for this purpose.

At the cost of some additional processing overhead, LDMS provides scale independence, automatic data consistency enforcement, and a high degree of configuration flexibility. One of the advantages of the LDS approach that should

now be apparent is that only data sets corresponding to frequently-referenced locational data need be kept on-line. Time series data, for example, could be easily stored in separate sets to facilitate both archival and comparison across the temporal dimension. Similarly, the master Region Dictionary could be archived and only an active subset of its entries kept on-line.

In summary, LDMS represents a fusion of previously-proven GIS and DBMS design elements which have been adapted to the LDS approach. Its contribution to GIS design development lies therefore in the strategy which it uses for associating data and location and its provisions for automatic enforcement of data consistency constraints.

References

1. J.C. Antenucci, "Design of the Geographic Information System Component of the Kentucky Natural Resource Information System ," pp. 9-20 in *Computer Mapping of Natural Resources and the Environment* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.

2. S.K. Chang, "A Methodology for Picture Indexing and Encoding," pp. 33-53 in *Picture Engineering*, ed. K.S. Fu and T.L. Kunii,Springer-Verlag, Berlin (1982).

3. M. Chock, A.F. Cardenas, and A. Klinger, "Manipulating Data Structures in Pictorial Information Systems," pp. 43-50 in *IEEE Computer*, (Nov 1981).

4. M. Chock, A.F. Cardenas, and A. Klinger, "Data Base Structure and Manipulation Capabilities of a Picture Data Management System (PICDMS)," pp. 484-492 in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (July 1984).

5. M.I. Chock, *A Data Base Management System for Image Processing*, University of California, Los Angeles (1982). Ph.D Dissertation

6. D. Comer, "The Ubiquitous B-Tree," pp. 121-137 in *Computing Surveys*, (June 1979).

7. J.T. Dalton, George E. Winkert , and John J. Quann , "A Raster-Encoded Polygon Data Structure for Interactive Mapping ," pp. 141-158 in *Cartographic and Statistical Data Bases and Mapping Software* , ed. Patricia A. Moore ,Laboratory for Computer Graphics and Spatial Analysis (1980). Volume 18 in Harvard Library of Computer Graphics/1980 Mapping Collection.

8. J. Dangermond, "A Classification of Software Components Commonly Used in Geographic Information Systems," *The Design and Implementation of Computer-Based Geographic Information Systems: Proceedings of a U.S./Australia Workshop at Honolulu, 1982*, IGU Commission on Geographical Data Sensing and Processing, (1984).

9. J. Dangermond, Logan Hardison , and Lowell K. Smith, "Some Trends in the Evolution of Geographic Information System Technology ," pp. 27-40 in *Computer Mapping of Natural Resources and the Environment* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.

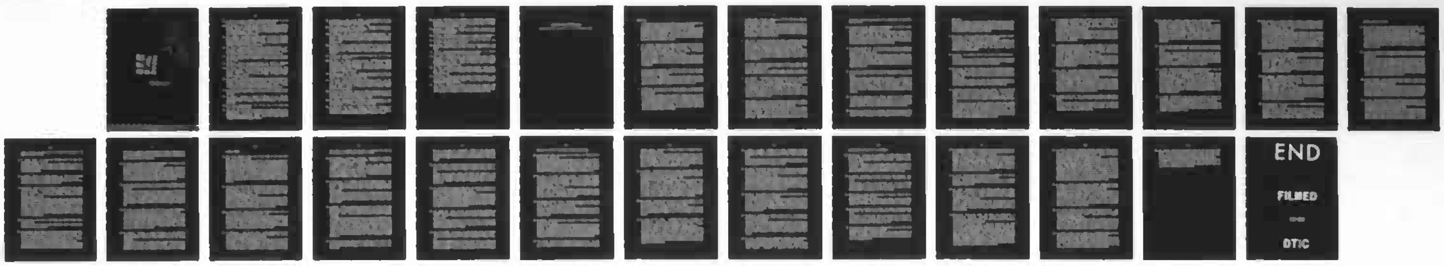
10. S.J. Eggers and A. Shoshani, "Efficient Access of Compressed Data," *Proc. 6th Intl. Conf. on Very Large Data Bases*, pp. 205-211 (Oct 1980).

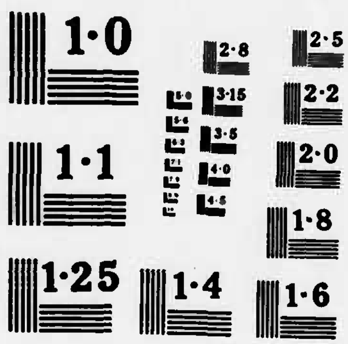
11. J.D. Foley and Andries van Dam, *Fundamentals of Interactive Computer Graphics*, Addison-Wesley (1983). Addison-Wesley Systems Programming Series.

12. A. Frank, "MAPQUERY: Data Base Query Language for Retrieval of Geometric Data and their Graphical Representation ," *Computer Graphics: SIGGRAPH '82 Conference Proceedings* 16 (3) pp. 199-207 (July 1982).

AD-A158 118 STRATEGIES FOR ASSOCIATING DATA AND LOCATION IN A GEOGRAPHIC INFORMATION SYSTEM(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH R G WAITON 1985 2/2

UNCLASSIFIED AFIT/CI/NR-85-78T F/G 8/2 NL





NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

13. A. Frank, "Application of DBMS to Land Information Systems," *Proc. 7th Intl. Conf. on Very Large Data Bases*, pp. 448-453 (Sept 1981).

14. R.V. Giddings, "A Computer System to Support Environmental Decision Making ," pp. 68-86 in *Urban, Regional, and State Government Applications of Computer Mapping* , ed. Patricia A. Moore ,Laboratory for Computer Graphics and Spatial Analysis (1980). Volume 11 in Harvard Library of Computer Graphics/1980 Mapping Collection.

15. D.D. Greenlee, "Application of Spatial Analysis Techniques to Remotely Sensed Images and Ancillary Geocoded Data ," pp. 111-120 in *Computer Mapping of Natural Resources and the Environment* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.

16. S.C. Guptill, "Thematic Map Production From Digital Spatial Data ," pp. 121-124 in *Computer Mapping of Natural Resources and the Environment* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.

17. A. Guzman, "Reconfigurable Geographic Data Bases," pp. 99-111 in *Pattern Recognition in Practice*, ed. E. Gelsema and L. Kanal,North-Holland Publishing Co. (1980).

18. A. Klinger, "Patterns and Search Statistics," pp. 303-337 in *Optimizing Methods in Statistics*, ed. J.S. Rustagi,Academic Press, New York (1971).

19. G.R. Koller, "Interpretation and Display of the NURE Data Base Using Computer Graphics ," pp. 41-50 in *Computer Mapping of Natural Resources and the Environment* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.

20. G.J. Langford, "Grid to Polygon Conversion in Geographic Information Processing: An Application to Resource Planning for the Cypress Hills ," pp. 51-56 in *Computer Mapping of Natural Resources and the Environment* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.

21. L. Lin, *The Physical Organization and Access Method for FICDMS--An Data Base Management System for Image Processing; M.S. Comprehensive Report.* July 1984.

22. G. Lohman, J. Stoltzfus, A. Benson, M. Martin, and A.F. Cardenas, "Remotely-Sensed Geophysical Databases: Experience and Implications for Generalized DBMS," *Proceedings, ACM SIGMOD Conference*, pp. 146-160 (May 1983).

23. D.B. Lomet, "Digital B-Trees," *Proc. 7th Intl. Conf. on Very Large Data Bases*, pp. 333-344 (Sept 1981).

24. J.H. Long, "The Importance of Documenting and Conserving Data in Cartographic Bases ," pp. 49-52 in *Cartographic and Statistical Data Bases and Mapping Software* , ed. Patricia A. Moore ,The Linkage of Data Describing Overlapping Geographical Units, (1980). Volume 18 in Harvard Library of Computer Graphics/1980 Mapping Collection.

25. T. Matsuyama, L. Hao, and M. Nagao, "A File Organization for Geographic Information Systems Based on the Spatial Proximity," *Proc. 6th Intl. Conf. on Pattern Recognition*, pp. 83-88 (Oct 1982).

26. D.M. McKeown, Jr. and Jerry L. Denlinger , "Graphical Tools for Interactive Image Interpretation ," *Computer Graphics: SIGGRAPH '82 Conference Proceedings* 16 (3) pp. 189-198 (July 1982).

27. J.D. McLaurin, "U.S. Geological Survey Digital Mapping Program ," pp. 53-59 in *Cartographic and Statistical Data Bases and Mapping Software* , ed. Patricia A. Moore ,Laboratory for Computer Graphics and Spatial Analysis (1980). Volume 18 in Harvard Library of Computer Graphics/1980 Mapping Collection.

28. C. Meade, "LUIS: An Interactive Graphics System Used for Data Base Management in a Regional Planning Environment ," pp. 119-126 in *Urban, Regional and State Applications* , Laboratory for Computer Graphics and Spatial Analysis (1979). Harvard Library of Computer Graphics/1979 Mapping Collection.

29. G. Nagy and S. Wagle, "Geographic Data Processing," pp. 139-181 in *Computing Surveys*, (June 1979).

30. N.L. Faust , L.E. Jordan , and M.D. Furman , "Development and Implementation of a Low-Cost Microcomputer System for LANDSAT Analysis and Geographic Data-Base Applications ," pp. 107-110 in *Computer Mapping of Natural Resources and the Environment* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.

31. P.E. Mantey and E.D. Carlson, "Integrated Geographic Data Bases: The GADS Experience ," pp. 173-198 in *Data Base Techniques for Pictorial Applications* , ed. A. Blaser ,Springer-Verlag (1980).

32. T.K. Peucker, "Literature for Geographic Information Systems ," pp. 175-179 in *Urban, Regional, and State Government Applications of Computer Mapping* , ed. Patricia A. Moore ,Laboratory for Computer Graphics and Spatial Analysis (1980). Volume 11 in Harvard Library of Computer Graphics/1980 Mapping Collection.

33. D. Peuquet, "Vector/Raster Options for Digital Cartographic Data," *The Design and Implementation of Computer-Based Geographic Information Systems: Proceedings of a U.S./Australia Workshop at Honolulu, 1982*, IGU Commission on Geographical Data Sensing and Processing, (1984).

34. R.L. Phillips, "Definition and Manipulation of Graphical Entities in Geographic Information Systems ," pp. 115-133 in *Data Base Techniques for Pictorial Applications* , ed. A. Blaser ,Springer-Verlag (1980).

35. D. Rhind and Tim Adams , "Coordinate Data Bases: Availability and Characteristics ," pp. 53-62 in *Cartographic Data Bases and Software* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 13 in Harvard Library of Computer Graphics/1981 Mapping Collection.

36. H. Samet, "Hierarchical Data Structures for Representing Geographical Information," *The Design and Implementation of Computer-Based Geographic Information Systems: Proceedings of a U.S./Australia Workshop at Honolulu, 1982*, ACM Computing Surveys., (1984).

37. H. Samet, A. Rosenfeld, C. Shaffer, and R. Webber, "Processing Geographic Data With Quadtrees," *Proc. 7th Intl. Conf. on Pattern Recognition*, pp. 212-215 (Jul-Aug 1984).

38. H. Samet, "Approximation and Compression of Images Using Quadtrees," *Proc. 7th Intl. Conf. on Pattern Recognition*, pp. 220-223 (Jul-Aug 1984).

39. H. Samet, "The Quadtree and Related Hierarchical Data Structures," pp. 187-260 in *Computing Surveys*, (June 1984).

40. M. Shneier, "Calculations of Geometric Properties Using Quadtrees ," pp. 296-301 in *Computer Graphics and Image Processing* , Academic Press (July 1981).

41. L. Tucker, "Model-Guided Segmentation Using Quadtrees," *Proc. 7th Intl. Conf. on Pattern Recognition*, pp. 216-219 (Jul-Aug 1984).

42. A.J. Ungar, "Getting More for Less--Polygon Graphics Using a Micro-Computer ," pp. 217-224 in *Cartographic and Statistical Data Bases and Mapping Software* , ed. Patricia A. Moore ,Laboratory for Computer Graphics and Spatial Analysis (1980). Volume 18 in Harvard Library of Computer Graphics/1980 Mapping Collection.

43. J.J. Utano, "A Portfolio of Computer Mapping Software at Akron University ," pp. 123-138 in *Cartographic Data Bases and Software* , locational data files. (1981). Volume 13 in Harvard Library of Computer Graphics/1981 Mapping Collection.

44. P.M. Wilson, "BASIS: The Bay Area Spatial Information System ," pp. 151-156 in *Urban, Regional and State Applications* , Laboratory for Computer Graphics and Spatial Analysis (1979). Harvard Library of Computer Graphics/1979 Mapping Collection.

45. A.L. Zobrist, "Integration of Landsat Image Data with Geographic Data Bases," *The Design and Implementation of Computer-Based Geographic Information Systems: Proceedings of a U.S./Australia Workshop at Honolulu, 1982*, IGU Commission on Geographical Data Sensing and Processing, (1984).

Appendix I: Annotated Bibliography

Strategies For Associating Data and Location in a Geographic Information System

References

1. T. Adams and K. Smith, *The Manipulation of Raster-Based Topographic Data on a Parallel Processor*, Manuscript, UCLA CSD Archives (1981).
The authors, from Birkbeck and Queen Mary Colleges in London, note the problems caused by collecting spatial data in raster form while manipulating it primarily in vector (topological) form. They describe a Distributed Array Processor (DAP) to perform the conversion using a parallel processing algorithm. The work has implications in the selection of data storage format for image and geographic information systems.

2. J.C. Antenucci, "Design of the Geographic Information System Component of the Kentucky Natural Resource Information System ," pp. 9-20 in *Computer Mapping of Natural Resources and the Environment* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.
Describes the design of a Geographic Information System oriented toward land use classification, especially as regards suitability for mining operations. The system resembles in many respects the Texas Natural Resource Information System (TNRIS) and the Minnesota Land Management Information System (MLMIS).

3. R. Berman and M. Stonebraker, "GEO-QUEL: A System for the Manipulation and Display of Geographic Data," pp. 186-191 in *Computer Graphics*, SIGGRAPH-ACM (Summer 1977).
A brief review of the design approach chosen for GEO-QUEL and a discussion of some of the problems encountered. These were chiefly performance related. Authors conclude that performance penalties inherent in a front-end design approach are offset by savings in software development costs.

4. F. Billingsley, "Data Base Systems for Remote Sensing," in *Data Base Techniques for Pictorial Applications*, ed. A. Blaser,Springer-Verlag, Heidelberg, Germany (1980).
Examines issues ranging from data acquisition, storage format, and archiving. Concludes that the preferred approach is to minimize processing of raw data but to append ancillary data to it for later use during retrieval processing. Identifies outstanding issues which must be addressed when designing geographic databases of national extent. The registered images suggested as a means of implementing an image based system are similar to the stacked image representation adopted by PICDMS.

5. A. Blaser (Ed.), *Data Base Techniques for Pictorial Applications*, Springer-Verlag, Heidelberg, Germany (1980).
Papers presented at an IBM-sponsored conference held in Florence, Italy in 1979. Contains a wealth of information on pictorial systems generally and approximately half of the papers deal with geographical information systems specifically. These address aspects ranging from requirements analysis to both software and hardware design. Several articles describe case studies, including GADS. An invaluable reference.

6. A.R. Boyle, "The Status of Graphic Data Input to Spatial Data Handling Systems," *The Design and Implementation of Computer-Based Geographic Information Systems: Proceedings of a U.S./Australia Workshop at Honolulu, 1982*, IGU Commission on Geographical Data Sensing and Processing, (1984).
Describes the types of processing equipment currently available for digitization of source data for spatial data information systems. The author sees a bright future for scanner input but concludes that manual line tracing and point digitization will continue to be important because, in many cases, it is more cost effective.

7. T.F. Brown, *Physical Organization and Access Strategies for Efficient Image Processing in PICDMS*, University of California, Los Angeles (Spring 1982). M.S. Comprehensive Report
Proposes enhancement of the PICDMS system through the addition of memory buffering mechanisms coupled with indexed sequential file structures

8. B.R. Burkhart, "Cartographic Base Files at Lawrence Berkely Laboratory: 1979 Inventory ," pp. 11-46 in *Cartographic Data Bases and Software* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 13 in Harvard Library of Computer Graphics/1981 Mapping Collection.
A comprehensive listing of the cartographic base files maintained by the Lawrence Berkely Laboratory, all of which are in the public domain. Virtually all of these reference regions of the U.S. and many are derived from census data. Descriptions for each file include its content, sources of data, currency, and a point of contact.

9. H.W. Calkins and R.F. Tomlinson, *Geographic Information Systems, Methods, and Equipment for Land Use Planning, 1977*, IGU Commission on Geographical Data Sensing and Processing, Ottawa, Canada (Aug 1977).
Report prepared by the International Geographic Union for the U.S. Department of the Interior Resource and Land Investigations (RALI) Program. Describes geographic information systems from the users point of view. Touches on a large number of design considerations, none of them in any great depth. Topics addressed include sources and characteristics of data,

manual and computer-assisted systems for its management, and representation alternatives.

10. N. Chang and K.S. Fu, "Query-by-Pictorial-Example," pp. 519-524 in *IEEE Transactions on Software Engineering*, (Nov 1980).
Describes an interface which extends Query-By-Example to support operations on databases of image data. The method involves development of application-specific processing packages to extract target data from the images. The information is then stored in the tables of a relational database. Entity representation is based on the topological model, but the article does not otherwise describe the access or data structures used.

11. N.S. Chang and K.S. Fu, "On the Construction of Relational Databases from Pictorial Data," *Proc. 1st Intl. Symp. on Policy Analysis and Information Systems*, pp. 128-138 (June 1979).
Description of the Relational Database System for Images (REDI) implemented at Purdue University. Describes the concept of a processing package to extract data from source images and the algorithms used to convert it to normalized tables in the database. Emphasis is on logical design; hardware, access structures, and the Query-By-Pictorial-Example user interface are not specifically addressed.

12. S.K. Chang, "A Methodology for Picture Indexing and Encoding," pp. 33-53 in *Picture Engineering*, ed. K.S. Fu and T.L. Kunii, Springer-Verlag, Berlin (1982).
Examines the subjects of picture indexing and abstraction in the context of a relational pictorial database. Proposes a methodology of representing pictures using structures similar to quadtrees but which incorporate a technique, called hypercube encoding, to reduce storage requirements. The method involves a compact representation, called a minimal enclosing hypercube, to represent a point set. Includes examples of query processing applied to these structures.

13. M. Chock, A.F. Cardenas, and A. Klinger, "Manipulating Data Structures in Pictorial Information Systems," pp. 43-50 in *IEEE Computer*, (Nov 1981).
An excellent introduction to the diversity of existing geographical information systems. Addresses the issues of logical image data structure and data manipulation operations as well as some of the limitations of current systems.

14. M. Chock, A.F. Cardenas, and A. Klinger, "Data Base Structure and Manipulation Capabilities of a Picture Data Management System (PICDMS)," pp. 484-492 in *IEEE Transactions on Pattern Analysis and Machine Intelligence*,

(July 1984).
Describes the database structure and data manipulation capabilities of a generalized picture database management system. The system is based on a dynamic stacked-image data structure, a generalization of the raster grid format. Includes extensive examples of the PICDMS data manipulation language and the flexibility of operations which it supports.

15. M.L. Chock, *A Data Base Management System for Image Processing*, University of California, Los Angeles (1982). Ph.D Dissertation
A thorough survey and comparison of existing computer cartographic and image processing systems and a description of Chock's own Pictorial Data Base Management System (PICDMS). Addresses the data structure, data manipulation language, application area, and performance aspects of PICDMS. Extensive bibliography.

16. D. Comer, "The Ubiquitous B-Tree," pp. 121-137 in *Computing Surveys*, (June 1979).
A classic tutorial covering B-Trees and the operations that may be performed on them. Some variations on the basic form are mentioned, but multidimensional generalizations (e.g. quadtrees) are not among them.

17. D.J. Cowen, "Using Standard Output Formats for Distributed Geographical Data Handling," *The Design and Implementation of Computer-Based Geographic Information Systems: Proceedings of a U.S./Australia Workshop at Honolulu, 1982*, IGU Commission on Geographical Data Sensing and Processing, (1984).
Identifies common elements in the output files of geographic information systems and provides examples of methods to transfer information between systems. Both grid and polygon-oriented representations are addressed.

18. J.T. Dalton, George E. Winkert , and John J. Quann , "A Raster-Encoded Polygon Data Structure for Interactive Mapping ," pp. 141-158 in *Cartographic and Statistical Data Bases and Mapping Software* , ed. Patricia A. Moore ,Laboratory for Computer Graphics and Spatial Analysis (1980). Volume 18 in Harvard Library of Computer Graphics/1980 Mapping Collection.
A functional overview of the Domestic Information Display System (DIDS) jointly developed by NASA and the US Bureau of the Census. Emphasis is on the hybrid raster/vector data representation used by the system. It is a modification of the run-length encoding scheme widely used for storage compression of data in raster format. The technique involves the addition of three additional fields, specifying distances to the nearest state and county boundaries, to the encoded definition of each raster line.

19. J. Dangermond, "A Classification of Software Components Commonly Used in Geographic Information Systems," *The Design and Implementation of Computer-Based Geographic Information Systems: Proceedings of a U.S./Australia Workshop at Honolulu, 1982*, IGU Commission on Geographical Data Sensing and Processing, (1984).
Describes various criteria by which geographic information systems may be categorized. These include the purposes for which they are used, the nature of the data and the operations performed on it, and the methods used to represent data. Emphasis is on establishing a reference framework; specific systems are not considered.

20. J. Dangermond, Logan Hardison , and Lowel K. Smith, "Some Trends in the Evolution of Geographic Information System Technology ," pp. 27-40 in *Computer Mapping of Natural Resources and the Environment* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.
Describes the implementation and usage trends of Geographic Information Systems, including the issues of data collection and integration, software and hardware development, and the extent of regions to be represented. Good general background but does not specifically address implementation issues or describe actual systems.

21. J.C. Davis, "Geographic Information Systems for Geologic Data," *The Design and Implementation of Computer-Based Geographic Information Systems: Proceedings of a U.S./Australia Workshop at Honolulu, 1982*, IGU Commission on Geographical Data Sensing and Processing, (1984).
The author identifies unsufficient recognition of the spatial nature of geological data as a problem in many current geological data management systems. Describes GIMMAP, a GIS jointly developed by the *Bureau de Recherches Geologiques et Minieres* of France and the Kansas Geological Survey which is especially tailored for geologic data. The system is based on graph-theoretic considerations and uses a topological data structure.

22. D. Edson and G. Lee, "Ways of Structuring Data Within a Digital Cartographic Data Base," in *Computer Graphics*, SIGGRAPH-ACM (Summer 1977).
Describes the data structures and file organizations developed to implement the U.S. Geological Survey's Digital Cartographic Database. Includes some discussion of the U.S. Census Bureau's Geographical Base File/Dual Independent Map Encoding (GBF/DIME) file.

23. S.J. Eggers and A. Shoshani, "Efficient Access of Compressed Data," *Proc. 6th Intl. Conf. on Very Large Data Bases*, pp. 205-211 (Oct 1980).
Describes a compression scheme suitable for use on large, stable statistical databases. The method involves a mapping between a source vector, which presumably includes many null or constant values, into a more compact header vector of counts. Algorithms to perform mappings between the source and header vectors are described and the method is compared with bit-map and run-length encoding schemes. The new method is judged to be superior, especially as regards access time: mapping in both directions is shown to be logarithmic.

24. J.D. Foley and Andries van Dam, *Fundamentals of Interactive Computer Graphics*, Addison-Wesley (1983). Addison-Wesley Systems Programming Series.
Introduction to computer graphics from the programmer's point of view. The material is presented in the framework of a vector-oriented Simple Graphics Package and stresses the use of efficient algorithms for manipulation and output. A considerable amount of the text deals with matrix transformation and shading of 3D objects. Also includes two chapters specifically on raster algorithms, software, and display architecture.

25. A. Frank, "Application of DBMS to Land Information Systems," *Proc. 7th Intl. Conf. on Very Large Data Bases*, pp. 448-453 (Sept 1981).
Describes the nature and implementation of a Land Information System (LIS) implemented over a general-purpose network model DBMS. A primary use of LIS systems is to retrieve maps interactively to display specified features and their surroundings. The project was implemented in Switzerland using DEC's DBMS-10 database manager to store information on land plots and cultural features. An additional quadtree structured file system was developed; both it and the access methods and search strategies used in query processing are described. Concludes that commercial DBMSs are suitable for LIS applications.

26. A. Frank, "MAPQUERY: Data Base Query Language for Retrieval of Geometric Data and their Graphical Representation ," *Computer Graphics: SIGGRAPH '82 Conference Proceedings* 16 (3) pp. 199-207 (July 1982).
An introduction to Land Information Systems and some of the problems involved in devising a suitable query language for use with them. Proposes a language, MAPQUERY, based on the popular SEQUEL query language and provides several examples of its use.

27. R.V. Giddings, "A Computer System to Support Environmental Decision Making ," pp. 68-86 in *Urban, Regional, and State Government Applications of Computer Mapping* , ed. Patricia A. Moore ,Laboratory for Computer Graphics and Spatial Analysis (1980). Volume 11 in Harvard Library of Computer Graphics/1980 Mapping Collection.
Describes a well-designed and integrated data management system developed for North Dakota's Regional Environmental Assessment Program (REAP). The system combines a conventional database management system, map and graphic display capability, and statistical interpretation of data. The DBMS serves as the central unifying component; it is used to manage polygon, grid, cellular, and alphanumeric data. Several standard commercial graphics and statistical packages, such as SPSS, DISSPLA, and TELL-A-GRAF are also integrated into the overall design.

28. A. Go, M. Stonebraker, and C. Williams, "An Approach to Implementing a Geo-Data System," *Proc. ACM SIGGRAPH/SIGMOD Workshop on Data Bases for Interactive Design*, pp. 67-77 (Sept 1975).
Describes the original design of the GEO-QUEL front end to the INGRES relational DBMS. Includes an overview of the basic INGRES query language, QUEL, and of tabular representation. Map relations are introduced as a special case of these general relational tables and additional commands to manipulate them are proposed. Addresses some of the implementation considerations in general terms.

29. D.D. Greenlee, "Application of Spatial Analysis Techniques to Remotely Sensed Images and Ancillary Geocoded Data ," pp. 111-120 in *Computer Mapping of Natural Resources and the Environment* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.
Describes techniques used by the Earth Resources Observation Systems (EROS) Data Center (EDC) to input and analyze geocoded data in conjunction with LANDSAT image data. Includes a description of the methods used to extract topographic data from standard cartographic source files. Ways of performing overlay, distance, and area analysis on gridded data and of creating raster images from point data are also discussed.

30. W. Greenup (Ed.), *Proc. of the Intl. Conf. on Computer-Assisted Cartography (AUTO-CARTO III)* . Jan 1978 .
A diverse collection of papers dealing with all aspects of computerized cartography and related issues. These range from general concerns such as economic requirements, data representation, data manipulation, and display technology to the specific design and capabilities of available hardware. Includes sections on raster-based data manipulation and on the role of

database management systems.

31. S.C. Guptill, "Thematic Map Production From Digital Spatial Data ," pp. 121-124 in *Computer Mapping of Natural Resources and the Environment* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.
Lists the major land use class categories used by the U.S. Geological Survey to develop a nationwide, multipurpose digital cartographic data base. Briefly describes the data encoding and map production techniques used. The underlying representational structure is topological/vector, although data is converted to a run-encoded raster format as a preliminary to production of actual printed maps.

32. A. Guzman, "Reconfigurable Geographic Data Bases," pp. 99-111 in *Pattern Recognition in Practice*, ed. E. Gelsema and L. Kanal,North-Holland Publishing Co. (1980).
Addresses the problem of balancing storage efficiency and processing efficiency when selecting a data representation model for geographic data. Suggests as a solution the storage of different categories of data in different formats, with adaptive conversion performed by the database as a function of usage patterns. This is feasible only if full information content is retained during conversions across formats. Inheritability of attributes is identified as an aid to reducing storage requirements, and some advantages of hybrid quadtree-topological representations are examined.

33. Harvard University , *Urban, Regional and State Applications* , Laboratory for Computer Graphics and Spatial Analysis (1979). Harvard Library of Computer Graphics/1979 Mapping Collection.
Includes papers on applications ranging from police and transit system planning to more general purpose systems developed by local, state, and regional governments. Among the systems described are the Maryland Automated Geographic Information System (MAGI) and the Bay Area Spatial Information System (BASIS), both of which are based on the grid structural model.

34. R. John, "Data Structure Considerations in Topographic Mapping," *The Design and Implementation of Computer-Based Geographic Information Systems: Proceedings of a U.S./Australia Workshop at Honolulu, 1982*, IGU Commission on Geographical Data Sensing and Processing, (1984).
Examines data structure requirements from the perspective of providing completeness and correctness in data representation. The conflicting requirement to provide efficient access to large amounts of data is discussed and some methods commonly used to balance these needs are described. One of the major problems considered is that of retrieving data relevant to a specific

display scale.This, the author asserts, is best solved by storing data on the basis of multi-scaled entities.

35. T. Joseph, *Higher Level Access for PICDMS; UCLA CS Dept, M.S. Comprehensive Report*. July 1984.
Describes a proposed query language for PICDMS which is similar to Query-By-Pictorial-Example but which includes additional extensions. The language allows queries to be framed in terms of either coordinate positions or named spatial entities.

36. A. Klinger, "Patterns and Search Statistics," pp. 303-337 in *Optimizing Methods in Statistics*, ed. J.S. Rustagi,Academic Press, New York (1971).
The paper that started the exploration of multidimensional search trees as a suitable access structure for spatially oriented data.

37. G.R. Koller, "Interpretation and Display of the NURE Data Base Using Computer Graphics ," pp. 41-50 in *Computer Mapping of Natural Resources and the Environment* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.
Describes the graphics and database system used by the Savannah River Laboratory (SRL) to store and analyze data gathered through geochemical analysis of water and sediment samples from 37 states. NURE refers to the National Uranium Resource Evaluation program established by the U.S. Department of Energy. The system integrates a number of different commercial graphics and statistical analysis packages with a collection of FORTRAN IV routines to manage a database of approximately 89,000 sample analyzes for each of 238 quadrangles.

38. K.S. Fu and T.L. Kunii (Eds.), *Picture Engineering*, Springer-Verlag, Berlin, Germany (1982).
Includes sections on pictorial database management, picture representation, and picture computer architecture.

39. G.J. Langford, "Grid to Polygon Conversion in Geographic Information Processing: An Application to Resource Planning for the Cypress Hills ," pp. 51-56 in *Computer Mapping of Natural Resources and the Environment* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.
A brief description of the POLYGRID general-purpose, hybrid geographic information system used for resource and land use planning in Cypress Hills Provincial Park, Alberta, Canada. The most attractive feature of the system

is its ability to represent and display data in either grid or polygon format. The grid to polygon conversion is performed in software by identifying the grid chains which form the map unit boundaries. The paper includes examples of equivalent maps in each format.

40. P. Larson, "Linear Hashing With Partial Expansions," *Proc. 6th Intl. Conf. on Very Large Data Bases*, pp. 224-232 (Oct 1980).
Reviews the basic concept of linear hashing and describes a modification which provides for more efficient file expansion. The technique involves performance of file expansions in increments. Additional memory is added to subsections of the file as necessary, rather than expanding the entire file at once. This produces higher average storage utilizations. Includes a comparative performance analysis between the original and revised schemes.

41. L. Lin, *The Physical Organization and Access Method for PICDMS--An Data Base Management System for Image Processing; M.S. Comprehensive Report.* July 1984.
A Comparison of the file structures originally implemented in PICDMS and some which have been proposed since. Proposes modification of the data dictionary to support a sub-image physical data organization and index file access method. The method is an implementation of the null-compression scheme suggested by Eggers and Shoshani for statistical databases. Discusses considerations in integrating PICDMS to the IBM 7350 Image Processing System.

42. W. Litwin, "Linear Hashing: A New Tool for File and Table Addressing," *Proc. 6th Intl. Conf. on Very Large Data Bases*, pp. 212-223 (Oct 1980).
A description and performance analysis of a new hashing scheme. The technique, termed linear virtual hashing, relies upon a dynamic address space and the use of a periodically-revised hashing function to reference it. Expansion of the address space is performed through a doubling algorithm which avoids the transfer of large numbers of records during revision. The attraction of the method lies in its ability to combine quick access with high memory load factors for dynamic files.

43. G. Lohman, J. Stoltzfus, A. Benson, M. Martin, and A.F. Cardenas, "Remotely-Sensed Geophysical Databases: Experience and Implications for Generalized DBMS," *Proceedings, ACM SIGMOD Conference*, pp. 146-160 (May 1983).
A discussion of the nature of geophysical data and some of the drawbacks in attempting to deal with it using conventional DBMSs. A prototype system developed at the Pasadena Jet Propulsion Laboratory is described. It couples videodisc image storage with descriptive data managed by the INGRES

relational DBMS.

44. D.B. Lomet, "Digital B-Trees," *Proc. 7th Intl. Conf. on Very Large Data Bases*, pp. 333-344 (Sept 1981).
The author describes a variation on the B-Tree structure which increases the fanout factor of each node through the use of a node doubling technique. Digital B-trees (DB-trees) distribute records among nodes composed of multiple physical pages. Page assignment is based on the leading binary digits of each key value, which permits identification of the proper page within a node without chain following. The increased fanout permits more rapid access to a required node and the record assignment algorithm allows immediate determination of the proper page within a node. Includes descriptions of the algorithms to operate on DB-trees and numerical analysis of their expected performance.

45. J.H. Long, "The Importance of Documenting and Conserving Data in Cartographic Bases ," pp. 49-52 in *Cartographic and Statistical Data Bases and Mapping Software* , ed. Patricia A. Moore ,Laboratory for Computer Graphics and Spatial Analysis (1980). Volume 18 in Harvard Library of Computer Graphics/1980 Mapping Collection.
Identifies some of the problems related to creation and maintenance of cartographic databases (CDB's). These include adequate documentation, provisions for update, and complications introduced by changes in region boundaries and collection of time-series data. One of the secondary references, *The Linkage of Data Describing Overlapping Geographical Units,*

46. J.L. Mannos (Ed.), *Design of Digital Image Processing Systems: Proc. of SPIE--The Intl. Soc. for Optical Eng., Aug 1981.* 1981.
Includes sections on both processing software and hardware components. As might be expected from the title, most of the papers deal with image processing and display rather than geographic and database issues per se. Notable exceptions are papers on "Image Processor Design Requirements in Land-Use Planning", "Geology and Image Processing", and "Digital Cartographic Systems at the Defense Mapping Agency Aerospace Center."

47. T. Matsuyama, L. Hao, and M. Nagao, "A File Organization for Geographic Information Systems Based on the Spatial Proximity," *Proc. 6th Intl. Conf. on Pattern Recognition*, pp. 83-88 (Oct 1982).
The authors introduce a k-dimensional binary search tree (k-d tree) which corresponds to a recursive partitioning of a two dimensional map space. In this regard, they are a further refinement quadtrees. The article describes the partitioning process and manipulation of line and region representations in some detail. Includes a comparative analysis of quadtrees and k-d trees,

from which it is concluded that the latter are preferable from both the storage requirement and retrieval efficiency standpoints.

48. D.M. McKeown, Jr. and Jerry L. Denlinger , "Graphical Tools for Interactive Image Interpretation ," *Computer Graphics: SIGGRAPH '82 Conference Proceedings* 16 (3) pp. 189-198 (July 1982).
A description of BROWSE, an interactive raster image display facility designed as the front end to a map image and photo-interpretation system (MAPS). The underlying design is based on a pyramidal hierarchy of multiple resolution images. Includes a description of the data structures and physical storage management strategy, as well as a comprehensive listing of the implemented commands and application areas.

49. J.D. McLaurin, "U.S. Geological Survey Digital Mapping Program ," pp. 53-59 in *Cartographic and Statistical Data Bases and Mapping Software* , ed. Patricia A. Moore ,Laboratory for Computer Graphics and Spatial Analysis (1980). Volume 18 in Harvard Library of Computer Graphics/1980 Mapping Collection.
A brief description of the goals of the U.S.G.S. National Mapping Program (NMP). Includes a list of the data categories to be included and the scales at which data is being collected for the various types of coverage. Several map figures showing the current and projected extent of those coverages accompany the paper.

50. C. Meade, "LUIS: An Interactive Graphics System Used for Data Base Management in a Regional Planning Environment ," pp. 119-126 in *Urban, Regional and State Applications* , Laboratory for Computer Graphics and Spatial Analysis (1979). Harvard Library of Computer Graphics/1979 Mapping Collection.
A description of the implementation and use of a Land Use Information System (LUIS) developed by the Greater Vancouver Regional District of British Columbia, Canada. LUIS incorporates several concepts which have direct counterparts or parallels in LDMS. These include separation of location and attribute data and the use of bounding rectangles to simplify both logical and output operations. Some of the major differences between the two include LUIS's use of the vector structural model and of physical rather than logical pointers to link locational and attribute data. LUIS is programmed in BASIC on a PDP/11.

51. S. Miller and S. Iyengar, "Representation of Regions of Map Data for Efficient Comparison and Retrievel," *Proc. IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition*, pp. 102-107 IEEE Comput. Soc. Press, (June 1983).

Proposes a topological data representation suitable for large geographical areas and high resolution. The method is a variant of run-length encoding and supports efficient comparison and retrieval operations. Includes data structures and algorithms for computing interactions between overlapping regions.

52. P.A. Moore (Ed.), *Computer Mapping of Natural Resources and the Environment,* Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.
Contains numerous papers dealing with specific GIS applications, with emphasis on the graphic interface aspects. Many of the systems are special purpose ones developed by government agencies ranging from the local to national level. Includes a large number of applications based on satellite-derived data.

53. P.A. Moore (Ed.), *Cartographic Data Bases and Software,* Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 13 in Harvard Library of Computer Graphics/1981 Mapping Collection.
Emphasis is on types and structures of available digital cartographic database files. These range from proprietary collections for sale by private companies to those maintained by government agencies such as the U.S. Bureau of the Census. The section on software is oriented toward automated map production methods; there are also several papers dealing with the subject of cadastral databases and land information systems.

54. P.A. Moore (Ed.), *Urban, Regional, and State Government Applications of Computer Mapping,* Laboratory for Computer Graphics and Spatial Analysis (1980). Volume 11 in Harvard Library of Computer Graphics/1980 Mapping Collection.
Most of the systems described are of statewide extent or represent initial implementations slated for expansion to that level. Includes a 16 page paper describing the development history and applications of the Texas Natural Resources Information System (TNRIS).

55. P.A. Moore (Ed.), *Cartographic and Statistical Data Bases and Mapping Software,* Laboratory for Computer Graphics and Spatial Analysis (1980). Volume 18 in Harvard Library of Computer Graphics/1980 Mapping Collection.
The section on mapping software includes a paper on the Areal Design and Planning Tool (ADAPT) system, used in Kentucky and Ohio on a statewide basis and in several other states to a more limited extent. A paper on Harvard's ODYSSEY system only briefly describes several of the available

programs, but includes a large number of example output maps.

56. G. Nagy and S. Wagle, "Geographic Data Processing," pp. 139-181 in *Computing Surveys*, (June 1979).
A survey on geographic data processing which includes a review of some of the characteristics of geographic data. Emphasis is on cartographic requirements and uses, although storage and retrieval aspects of descriptive data is also addressed. Includes sections describing various storage formats and the processing operations used with them. Describes the design and available operations of ten specific systems, including DIME, CGIS, and GADS.

57. N.L. Faust , L.E. Jordan , and M.D. Furman , "Development and Implementation of a Low-Cost Microcomputer System for LANDSAT Analysis and Geographic Data-Base Applications ," pp. 107-110 in *Computer Mapping of Natural Resources and the Environment* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 15 in Harvard Library of Computer Graphics/1981 Mapping Collection.
Provides a brief overview of the NIMGRID microcomputer-based spatial analysis system derived from the earlier GRID and IMGRID systems developed at Harvard. NIMGRID is a Z-80 based S-100 system which uses the raster-oriented version of the grid data model.

58. P.E. Mantey and E.D. Carlson, "Integrated Geographic Data Bases: The GADS Experience ," pp. 173-198 in *Data Base Techniques for Pictorial Applications* , ed. A. Blaser ,Springer-Verlag (1980).
Overview of the system architecture, data manipulation language, and implementation of IBM's Geo-data Analysis and Display System. GADS is an interactive system based on the relational data model and polygon/topological structural model.

59. T. Peucker, *Computer Cartography*, Association of American Geographers, Washington,D.C. (1972).
Focus is upon cartographic requirements and operations; much of the work is therefore too application-specific to apply to geographic information systems generally. Addresses such issues as map projections, surface representation, and correct transformations of the base data to produce various cartographic products. The section on data structures and data organization discusses data encoding and basic storage options in a somewhat more general fashion.

60. T.K. Peucker, "Literature for Geographic Information Systems ," pp. 175-179 in *Urban, Regional, and State Government Applications of Computer Mapping* , ed. Patricia A. Moore ,Laboratory for Computer Graphics and Spatial

Analysis (1980). Volume 11 in Harvard Library of Computer Graphics/1980 Mapping Collection.
Much more than just a listing of major papers and texts relating to Geographic Information Systems, although such a list is included. The paper also traces developments in the field, identifying disciplines and individuals who have made significant contributions. The author does not hesitate to note what he considers to be mistakes, wrong turns, and missed opportunities along the way. An excellent short introduction to the subject.

61. D. Peuquet, "Vector/Raster Options for Digital Cartographic Data," *The Design and Implementation of Computer-Based Geographic Information Systems: Proceedings of a U.S./Australia Workshop at Honolulu, 1982*, IGU Commission on Geographical Data Sensing and Processing, (1984).
Identifies the problems inherent in collecting source data in raster form and converting it to vector (topological) format for manipulation. This discrepancy is largely a result of data acquisition by remote sensing or conversion from archival sources using mass digitization. Data output must also often use the raster format. In many cases, however, efficient algorithms exist only for manipulating data in topological form. Existing methods for dealing with this dilemma are discussed and an alternative method, conversion of source data to a hybrid "VASTER" form, is proposed.

62. R.L. Phillips, "Definition and Manipulation of Graphical Entities in Geographic Information Systems ," pp. 115-133 in *Data Base Techniques for Pictorial Applications* , ed. A. Blaser ,Springer-Verlag (1980).
Explores the graphic requirements of two geographic information systems, one a cartographic system dealing with oil leases and production, the other a water quality database (STORET) used by the Environmental Protection Agency.

63. D. Rhind and Tim Adams , "Coordinate Data Bases: Availability and Characteristics ," pp. 53-62 in *Cartographic Data Bases and Software* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 13 in Harvard Library of Computer Graphics/1981 Mapping Collection.
An excellent discussion of the data resolution and representation problem from the perspective of storage requirements. Includes descriptions of the resolution, types, and numbers of features stored by several vector and grid/raster based systems. Identifies some of the past problems in accurately estimating and dealing with the volume of data generated by a selected resolution level. Notes some recent trends in data collection and processing technology and their implications.

64. L. Salmon, J. Gropper, J. Hamill, and C. Reed, "Comparison of Selected Operational Capabilities of Fifty-Four Geographic Information Systems," FWS/OBS-77/54, Fish and Wildlife Service, U.S. Dept. of the Interior (Sept 1977).
Identifies 85 systems and compares 54 of the systems on the basis of software and hardware characteristics, as well as the types of tasks performed. The systems themselves are not described in any detail.

65. H. Samet, "Hierarchical Data Structures for Representing Geographical Information," *The Design and Implementation of Computer-Based Geographic Information Systems: Proceedings of a U.S./Australia Workshop at Honolulu, 1982*, ACM Computing Surveys., (1984).
Describes the structure of quadtrees and their use to support operations commonly performed by geographic information systems. Emphasis is on basic quadtrees, although the strip tree variation for representing line data is also described at some length. Other variations are referenced but not developed. Much of the same information is presented by the author in greatly expanded form in the June 84 issue of *ACM Computing Surveys*.

66. H. Samet, "The Quadtree and Related Hierarchical Data Structures," pp. 187-260 in *Computing Surveys*, (June 1984).
A comprehensive tutorial survey of quadtrees and a multitude of variations on them. This is the best source of information on these data structures and the types of operations they support. Includes an extensive reference list of recent papers, many of them specifically addressing the subject of geographic information system design. Consolidates much of the author's previous work into a single reference.

67. H. Samet, A. Rosenfeld, C. Shaffer, and R. Webber, "Processing Geographic Data With Quadtrees," *Proc. 7th Intl. Conf. on Pattern Recognition*, pp. 212-215 (Jul-Aug 1984).
Describes a quadtree-based memory management scheme in which space is decomposed into a hierarchy of equal-sized parts. Data is stored in the quadtree leaves, and these are organized linearly without the use of explicit pointers. Individual leaves are referenced by a B-tree index maintained on disk. A database language is also defined to specify operations on the resulting files.

68. H. Samet, "Approximation and Compression of Images Using Quadtrees," *Proc. 7th Intl. Conf. on Pattern Recognition*, pp. 220-223 (Jul-Aug 1984).
Description of a quadtree variation which does not require space for storage of explicit pointers between nodes; the tree is stored as a collection of leaf nodes. Includes a description of algorithms suitable for extracting

approximations of images from these structures.

69. W. Sharpley, J. Leiserson, and A. Schmidt, "A Unified Approach to Mapping, Charting and Geodesy (MC&G) Data Base Structure Design," ETL-0144, U.S. Army Engineer Topographic Laboratories, Analytical Sciences Corp. (May 1978).
Analyzes the implications of various image archive structures and media in some depth. Describes the characteristics of topological data formats and operations commonly performed on them. Discusses the design of an image data base built upon the capabilities of ODYSSEY, an existing system.

70. M. Shneier, "Calculations of Geometric Properties Using Quadtrees ," pp. 296-301 in *Computer Graphics and Image Processing* , Academic Press (July 1981).
A presentation in Pascal of the algorithms to be used for computing geometric properties of binary images represented as quadtrees. These include algorithms to find the area, centroid, intersection, union, and complement of binary images using a tree-traversal strategy.

71. IEEE Computer Society, *IEEE Transactions on Computers (Special Issue on Image Database Mgmnt).* Oct 1982.
Special Issue on Computer Architecture for Pattern Analysis and Image Database Management. Contains considerable discussion of hardware and design issues and includes articles on the PUMPS and PICCOLO prototypes.

72. M. Stonebraker, E. Wong, and P. Kreps, "The Design and Implementation of INGRES," pp. 189-222 in *ACM Transactions on Database Systems*, (Sept 1976).
A description of INGRES itself, without specific reference to GEO-QUEL. The section on data structures and access methods is fairly detailed. It includes the intuitive logic behind most of the design choices in those areas but no formal calculations of the tradeoffs involved.

73. M. Stonebraker, "Retrospection on a Database System," pp. 225-240 in *ACM Transactions on Database Systems*, (June 1980).
Reviews the evolution and design decisions made during development of INGRES, with emphasis upon the mistakes made and lessons learned. Performance of GEO-QUEL is directly impacted by many of those design flaws, but is not addressed directly.

74. R.F. Tomlinson, "Difficulties Inherent in Organizing Earth Data in a Storage Form Suitable for Query ," *Proc. of the Intl. Conf. on Computer-Assisted Cartography (AUTO-CARTO III)* , pp. 181-201 (Jan 1978).
Identifies some of the problems inherent in organizing earth data in a form suitable for efficient query. Among the issues raised are the difficulties in dealing with the rapidly growing volume of digital data and the nature of spatial relationships. Tradeoffs involved in selecting which relationships might be best derived on demand and which should be explicitly stored are noted, and the suitability of the three conventional DBMS data models for representing those relationships is considered.

75. L. Tucker, "Model-Guided Segmentation Using Quadtrees," *Proc. 7th Intl. Conf. on Pattern Recognition*, pp. 216-219 (Jul-Aug 1984).
Describes a quadtree implementation scheme in the context of a medical imagery storage and analysis system. Some of its features have application to geographic systems as well.

76. A.J. Ungar, "Getting More for Less--Polygon Graphics Using a Micro-Computer ," pp. 217-224 in *Cartographic and Statistical Data Bases and Mapping Software* , ed. Patricia A. Moore ,Laboratory for Computer Graphics and Spatial Analysis (1980). Volume 18 in Harvard Library of Computer Graphics/1980 Mapping Collection.
Describes a system for creating thematic maps and doing spatial analysis based on polygon graphics. Known as SHADE (Simple Handling of Areal Data Expressions), the design features the use of bounding rectangles to screen polygons, a multi-level tree-structured polygon file system, and map overlay capability.

77. International Geographical Union, *The Design and Implementation of Computer-Based Geographic Information Systems: Proceedings of a U.S./Australia Workshop at Honolulu, 1982*, IGU Commission on Geographical Data Sensing and Processing, (1984).
Also available as Peuquet, Donna J., and John O'Callaghan, eds. 1983. Proceedings, United States/Australia Workshop on Design and Implementation of Computer-Based Geographic Information Systems. Amherst, NY: IGU Commission on Geographical Data Sensing and Processing. One of the most current and comprehensive sources of information on Geographical Information Systems. Includes sections on Computer Generated Cartographic Displays, Organization and Management of Very Large Spatial Data Bases, Creation and Integration of Spatial Data Bases, Design of large Spatial Decision Support Systems, and Applications of Geographic Information Systems. Most papers provide many additional, recent references.

78. J.J. Utano, "A Portfolio of Computer Mapping Software at Akron University ," pp. 123-138 in *Cartographic Data Bases and Software* , Laboratory for Computer Graphics and Spatial Analysis (1981). Volume 13 in Harvard Library of Computer Graphics/1981 Mapping Collection.
A fairly detailed description of the statistical mapping software (MAPSOFT) developed at Akron University and numerous examples of the maps produced. Includes diagrams and a discussion of the overall structure of the software, which emphasizes modular design methods, as well as a further description of the major software components. The programs are written in FORTRAN and the final output is by drum plotter. The system uses a vector representation to produce files defining the outlines of polygon regions or collections of point locations. These are known as *locational data files*. Each locational data file is paired with a corresponding statistical data description file containing attribute data associated with the regions or points.

79. P. Vaidya, L. Shapiro, R. Haralick, and G. Minden, "Design and Architectural Implications of a Spatial Information System," pp. 1025-1031 in *IEEE Transactions on Computers*, IEEE Computer Society (Oct 1982).
An examination of the issues, development, and design considerations in implementing spatial information systems. Describes implementation of an entity-oriented spatial database system based on the relational data model and the topological structural model. Includes a description of the data structures used at the entity, logical, internal memory, and external storage levels of the system. Addresses some processing efficiency issues relating to architecture and memory access strategy.

80. P.M. Wilson, "BASIS: The Bay Area Spatial Information System ," pp. 151-156 in *Urban, Regional and State Applications* , Laboratory for Computer Graphics and Spatial Analysis (1979). Harvard Library of Computer Graphics/1979 Mapping Collection.
A description of the design constraints and features of BASIS and the uses to which it is being put. BASIS is a grid cell system which uses logical cells of hectare size, aggregated into 10 X 10 hectare arrays for physical storage on a square kilometer basis. System extent is approximately 1000 square miles, with a storage capacity of 80 data items for each of its two million grid cells.

81. A. Zobrist, N. Bryant, and A. Landini, *Use of LANDSAT Imagery for Urban Analysis*, Manuscript, UCLA Computer Science Archives (1977).
Discusses the grid, polygon, and raster structures and concludes that raster is the representation of choice. Describes several examples of urban land use applications, including crosstabulation, estimation, and projection of current data. The ability of LANDSAT imagery to provide the requisite data, and its storage and manipulation in raster form, are recurring themes.

82. A.L. Zobrist, "Integration of Landsat Image Data with Geographic Data Bases," *The Design and Implementation of Computer-Based Geographic Information Systems: Proceedings of a U.S./Australia Workshop at Honolulu, 1982*, IGU Commission on Geographical Data Sensing and Processing, (1984).
Describes some of the basic components of the Image Based Information System (IBIS). This system stores data in raster, vector, and tabular forms. In the course of processing, data is frequently converted between them and these transformation methods are described.

END

FILMED

10-85

DTIC